Boundary Finding with Parametrically Deformable Models

Lawrence H. Staib, *Member, IEEE*, and James S. Duncan, *Senior Member, IEEE*

Abstract— **Segmentation using boundary finding is enhanced both by considering the boundary as a whole and by using model-based global shape information. Previous boundary finding methods have either not used global shape or have designed individual shape models specific to particular shapes. We apply flexible constraints, in the form of a probabilistic deformable model, to the problem of segmenting natural 2-D objects whose diversity and irregularity of shape make them poorly represented in terms of fixed features or form. The parametric model is based on the elliptic Fourier decomposition of the boundary. Probability distributions on the parameters of the representation bias the model to a particular overall shape while allowing for deformations. Boundary finding is formulated as an optimization problem using a maximum *a posteriori* objective function. Results of the method applied to real and synthetic images are presented, including an evaluation of the dependence of the method on prior information and image quality.**

Index Terms— **Boundary finding, deformable models, Fourier descriptors, parametric models, probabilistic models, 2-D representation and recognition.**

I. Introduction

A POWERFUL property for distinguishing an object from its surroundings in an image is overall shape. Shape can be used to complete the information provided by local properties of the image such as gray level, texture, or color. A preconception of local shape features, such as curvature, can be useful, but these features are not as expressive as global shape and, like other local features, are more sensitive to poor viewing conditions. Global shape is too varied to be adequately described by a single shape attribute such as average bending energy or compactness. In order to take full advantage of shape, the problem of object identification will be approached as a process of boundary finding or delineation using a boundary measure and incorporating global shape information.

Segmentation by boundary finding using only local information has often been frustrated by poor-contrast boundary regions due to occluding and occluded objects, adverse viewing conditions, and noise. A model-free interpretation is doomed by the underconstrained nature of the problem. Imperfect image data can be augmented with the extrinsic information that a geometric shape model provides. In order to exploit model-based information to the fullest extent, it should be incorporated explicitly, specifically, and early in the analysis. Some models incorporate generic information such as smoothness or low overall curvature. Although this is completely appropriate when no better information is available, it is better if the shape information used is as specific as possible. Applying the model too late allows inconsistencies to be created by the low-level processing. In addition, the boundary can be profitably considered as a whole because it tends to result in a more overall consistent solution.

This work is aimed at segmenting 2-D objects from 2-D images. It is especially focused on natural objects, such as those found in biomedical images, whose diversity and irregularity of shape make them poorly represented in terms of fixed features or form. The objects, such as organs, cells, and other biological structures, are expected to have a tendency toward some average shape with a continuum of possibilities near that average shape. This tendency can be taken advantage of by its expression in an appropriately designed shape model. An object's appearance in an image is governed by both its shape and by the view of the object. The view determines the translation, rotation, and scale of the object. The view of an object is likely to be variable unless it is explicitly controlled. In this work, the view will be taken as approximately known or determined by some preprocessing. Biomedical images are an important application because automatic quantitative analysis of objects present in these images is needed for both research and clinical purposes. These images can have the usual problems of noise, occlusions, and poor contrast. In addition, they provide a rich domain for the study of shape. This model domain lies between the extremes of a completely arbitrary object and a fixed template. A complete system using model-based optimization for the determination of object boundaries has been developed [33].

The next section discusses alternate methodologies in boundary finding. Section III discusses parametrization alternatives and then develops the Fourier parametrization to model closed and open curves. Section IV develops the objective function for fitting the boundary model to the image. Section V explains the optimization methods and implementations used. Section VI describes experiments showing the evaluation of this method on real and synthetic images.

II. Related Work in Boundary Finding

To some, using boundary methods means doing edge detection, that is, calculating a binary edge image. The problem

Manuscript received September 27, 1990; revised August 5, 1991. This work was supported by the National Institutes of Health under grant T15LM07056 (National Library of Medicine) and grant R29HL38333 (National Heart, Lung, and Blood Institute). Recommended for acceptance by Associate Editor C. Brown.

The authors are with the Departments of Diagnostic Radiology and Electrical Engineering, Yale University, New Haven, CT 06520.

IEEE Log Number 9202384.

with edge detectors for boundary finding is that the edges found do not necessarily correspond to boundaries of objects. With the exception of high-quality images from controlled environments, edge detectors produce spurious edges and gaps. Thus, although boundary information is undoubtedly useful, edge detectors *per se* are of limited use in general and of no use in poor-quality images. The limitations of edge detectors are due in part to their complete reliance on computations made directly on a local neighborhood of pixels in the raw image. This ignores both model-based information and higher order organization of the image. The methods most closely related to the one presented in this paper are discussed below, including grouping, pixel search, Hough methods, and whole-boundary methods that are optimized in either a parameter space or in the image space.

Grouping is one way of associating edge elements for the purpose of determining boundaries [21], [25]. The association is done using similarity relations, perhaps in conjunction with model information. Forming a complete boundary is usually accomplished by first associating individual edge elements into edge segments and then associating the segments into a boundary. Grouping methods, although robust to areas of weak boundary definition, often resort to arbitrary interpolation in order to form a complete boundary. In addition, it is often difficult to identify and discount spurious edge segments.

Pixel search methods attempt to find an optimal path through an image based on criteria designed to find boundaries. The typical objective function combines boundary strength and low overall curvature. Dynamic programming [26] and heuristic search [23] have been used to solve the optimization problem.

An alternative method for boundary analysis is the Hough transform [4], [10]. The Hough approach is similar to the current method in that it finds shapes by looking for maxima in a parameter space and is based on template matching. However, the storage and computational complexity of the Hough method are a great disadvantage, especially if deformations are envisaged. The advantage of the current technique, as will be seen, is that the entire parameter space does not have to be constructed due to the use of local search in finding these maxima.

Other investigators have considered whole-boundary methods that adjust a tentative boundary in order to find a boundary in an image. By considering the boundary as a whole, a structure is imposed on the problem that simplifies the task. Gaps are easily bridged, and overall consistency is more likely to result. The method described in this work is a whole-boundary method. One of the first instances of this type of approach is that of Widrow [34], who used parametrized templates called rubber masks to model objects. The parameters are sizes and relationships between subparts. Yuille *et al.* [36] used a parametrized template for an eye consisting of a circle bounded by two parabolas. The template was deformed to the image by optimizing a cost function based on morphological features. They developed a similar template for the mouth. Both of these methods have the advantage of describing the overall shape of the structure using very few parameters. However, the object must have sufficient structure to be represented in terms of parts, and a new model must be developed for each new object. Work has also been done developing deformable templates based on Markov models of 2-D boundaries incorporating knowledge of shape from statistical features [8]. These part models are related to the representation used by Fischler and Elschlager [12], where components of an object are held together by spring forces. The current method does not rely on a part description and is not specific to a particular object but supplies specific global shape information. Schudy [5] used a spherical harmonic parametrization for boundary finding. This model is fairly general but provides no explicit shape information other than smoothness. All of these methods fit the model to the image data by searching the parameter space for the best fit.

Although the above whole-boundary methods optimized in parameter spaces, the following methods optimize in the image space. Gritton and Parrish [15] used a flexible bead chain, where the beads are putative boundary points. The beads are attracted toward pixels that have a higher gradient magnitude. Cooper [9] formulated boundary estimation using maximum likelihood. A boundary adjustment scheme similar to the bead chain algorithm [15] is presented to perform the optimization. Kass *et al.* [18] used energy-minimizing snakes that are attracted to image features such as lines and edges, whereas internal spline forces impose a smoothness constraint. The weights of the smoothness and image force terms in the energy functional can be adjusted for different behavior. The solution is found using variational methods. The use of image space representations for the boundary (as opposed to parameter space) makes it difficult to incorporate global shape information, and none of these methods do so. These boundary finding methods are related to elastic matching methods used for the similar problem of registering images [2], [7].

All of these whole-boundary approaches take an initial estimate of the contour and adjust it to optimize some measure of fit. Widrow and Yuille *et al.* used explicit global shape information. Their models are best designed for structures with well-defined parts. The other methods described limit their use of shape information to overall smoothness properties and the implicit shape information provided by the initial placement of the contour. Prior information can range from the very general (e.g., smoothness constraints) to the very specific (e.g., exact templates). The boundary finding method described below is aimed at the situation between the extremes, where there is some prior information about the global shape of the object, but it is not exact.

III. Model

The objects being modeled have smooth boundaries that are continuously deformable with no obvious decomposition. Because overall shape is the only reliable salient feature, a uniform representation that describes the entire shape is needed. The prior information available is a flexible bias toward more likely shapes. This sort of model can be achieved by using a generic parametrization with probability distributions defined on the parameters, that is, the parametrization itself will be expressive enough to represent any potential shape of a given geometric type (for example, closed curves),

but the associated probability distributions will introduce a bias toward an expected range of shapes. The spread in the distributions is due to variability among instances of the object. This kind of parametrization represents a stronger use of prior information than methods that use only simple shape characteristics.

This section first discusses general design considerations for the parametrization and briefly mentions other possible methods. The elliptic Fourier method is then described for both closed and open curves, including a derivation of a geometric interpretation of the parametrization as a decomposition into ellipses. The number of harmonics used is then discussed. This section concludes with an explanation of the probability distributions that are associated with the parameters of the representation.

A. Parametrization Design Considerations

Each parametrization has particular properties that suit it for different purposes. The class of shapes that a parametrization can express is important because it represents a limitation of domain. Some restrictions, such as smoothness, can be convenient because they build a necessary constraint directly into the representation. Other restrictions, such as convexity, could represent a design compromise in that the class of problems addressed is limited by the representation. For fitting purposes, it is important that there is a one-to-one, continuous mapping between the shape and its parametrization. This allows fitting in the parameter space. In addition, it is desirable for the parametrization to be concise because that determines the complexity of the optimization process.

B. Alternative Parametrizations

Many boundary representations that are potentially useful for fitting have been developed [6]. Direct representations are an explicit list of the coordinates representing the object, which is usually indexed by a spatial parameter. Any curve, for example, can be ordered by arclength s and represented as $x(s)$ and $y(s)$. It can be advantageous to represent a differential property of the curve, using, for example, chain codes (discrete tangent), $\psi(s)$ (continuous tangent) [5], or the extended circular image (curvature) [17]. Differential properties allow rotation and scale changes easily. The extended circular image is the 2-D analog to the extended Gaussian image, but it requires convex curves or curve segments. Interpolating splines are piecewise polynomials with specifiable continuity properties that interpolate between a set of control points. Splines and other piecewise representations are essentially direct representations that are restricted to certain desired functional forms. Superellipses [3] are an extension of ellipses using an exponent that varies the shape continuously from elliptical to rectangular. The basic shape can be altered by such operations as twisting, bending, and tapering, as can any explicit representation. Even with these altering operations, superellipses are limited by their symmetric cross section and, thus, still only represent a very limited family of shapes (without resorting to composition).

C. Fourier Parametrizations

Fourier representations are those that express the curve in terms of an orthonormal basis. The motivation for a basis representation is that it allows us to express any object as a weighted sum of a set of known functions. An orthonormal set is desirable because it makes the parameters distinct. This makes coefficient determination easier and avoids redundancy.

To express the function $X(t)$ on the interval (a, b) in terms of the basis $\phi_k(t)$, we write

$$X(t) = \sum_{k=1}^{\infty} p_k \phi_k(t) \qquad \text{where} \qquad p_k = \int_a^b X(t)\phi_k(t)\,dt. \tag{1}$$

The coefficients p, which are the projections of the function onto the k basis functions, are the parameters of the representation. In order to use this representation, however, the sum must be truncated. In most such representations, the higher indexed basis functions represent higher spatial variation. Therefore, if the function to be represented is expected to have limited spatial variation, as is the case for most real object boundaries, the series can be truncated and still accurately represent the function.

The usual basis functions are the sinusoids [27], although others, such as those based on orthogonal polynomials, are possible. The sinusoids have the advantage of representing the familiar notion of frequency. The use of Fourier representations of contours has been limited primarily to classification applications such as character recognition [14] and airplane silhouettes [24]. The various representations of curves usually differ in the choice of direct representation on which to base the decomposition, such as $r(\theta)$, $\psi(s)$, $[x(s), y(s)]$, and $x(s)+iy(s)$. The $r(\theta)$ representation limits the curves to radial ones. $\psi(s)$ is not a good Fourier representation because it is differential. Curves reconstructed from a truncated series for a closed curve using the $\psi(s)$ representation may not be closed. Using a differential representation also exacerbates the problem of corners because they become discontinuities. The $(x(s), y(s))$ and $x(s) + iy(s)$ representations are completely expressive and do not have the problems of differential representations.

Fourier parametrizations are the most suitable for this work for a number of reasons. They are concise and not limited to a particular class of objects. The conversion between the parametrization and the shape is easily and directly computable due to the fast Fourier transform. By using a truncated series, we limit the frequency content of the curve.

D. Closed Curves

The standard real Fourier representation is based on (1), using the sinusoids or trigonometric functions as the basis functions, that is

$$\phi = \left\{ \frac{1}{2\pi}, \frac{\cos x}{\pi}, \frac{\sin x}{\pi}, \frac{\cos 2x}{\pi}, \frac{\sin 2x}{\pi}, \ldots \right\}. \tag{2}$$

These can be used, in conjunction with a direct representation, as the basis for parametrizations of closed curves. Closed curves are useful for representing organs, cells, and other

objects that are delineated by a complete boundary. A closed curve can be represented by two periodic functions of t, where t varies from 0 to 2π, $x(t)$, and $y(t)$. If we then take the Fourier decomposition of these two functions using the sinusoidal basis and write the resulting equations in matrix form, we get the elliptic Fourier representation [13], [19], [20]

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \tag{3}$$

where

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} x(t)dt \qquad c_0 = \frac{1}{2\pi}\int_0^{2\pi} y(t)dt$$

$$a_k = \frac{1}{\pi}\int_0^{2\pi} x(t)\cos kt\, dt \qquad b_k = \frac{1}{\pi}\int_0^{2\pi} x(t)\sin kt\, dt$$

$$c_k = \frac{1}{\pi}\int_0^{2\pi} y(t)\cos kt\, dt \qquad d_k = \frac{1}{\pi}\int_0^{2\pi} y(t)\sin kt\, dt.$$

The closed curve is thus represented by $\boldsymbol{p}_{\text{raw}} = (a_0, c_0, a_1, b_1, c_1, d_1, \ldots)$, which will be referred to as the raw parameter vector. This particular version of Fourier boundary representation has a number of advantages. A geometric interpretation, in terms of ellipses, can be developed from this decomposition. The geometric interpretation will allow for better visualization of the effect of the parameters and invariance to starting point, scale, and 2-D rotation and translation. Invariance to rotation, scale, and translation is important because these parameters are determined not by the object but by the view of the object, which often cannot be held constant.

In (3), the first two coefficients a_0 and c_0 determine the overall translation of the shape. Each term in the summation is the parametric form for an ellipse. In the degenerate case, $a_k d_k - b_k c_k = 0$ and the parametric form defines a straight line (a degenerate ellipse). In each term, the matrix determines the characteristics of the ellipse. The contour can be viewed as being decomposed into a sum of rotating phasors, each individually defining an ellipse and rotating with a speed proportional to their harmonic number k. This can be seen in Fig. 1, where a contour is shown constructed from three component ellipses forming a sort of planetary system. The straight lines represent the phasors for each ellipse shown at three different times. Thus, the point C_{ij} traces out the ith ellipse at time j. Each point is the center of the next higher ellipse. C_0 is the center of the first ellipse. Points C_{31}, C_{32}, and C_{33} are three different points on the final curve.

It is important that the curve representation that is decomposed into Fourier components be both continuous and periodic. Discontinuities slow the convergence because of the high frequencies inherent in a step jump. In this representation, both $x(t)$ and $y(t)$ are periodic because the contour is closed, and both $x(t)$ and $y(t)$ are continuous because the contour is continuous.

In (3), we can make t correspond to arclength by taking $t(s) = \frac{2\pi s}{S}$, where s is arclength along the curve from the starting point, and S is the total arclength of the curve. However, when a curve is reconstructed from a truncated series, the relationship no longer holds. This can be seen by noting

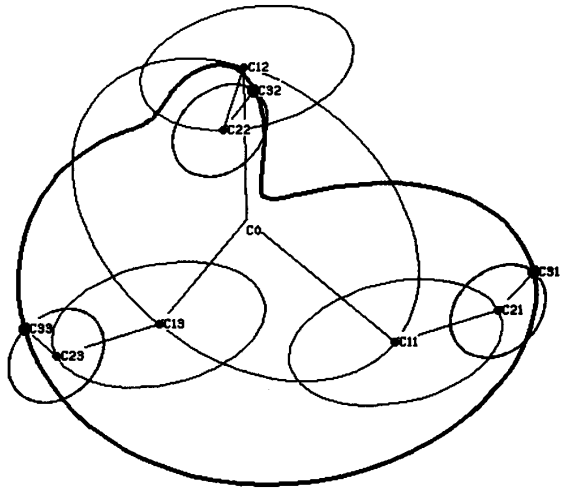

Fig. 1. Contour (dark line) at the left is constructed from three component ellipses shown at three different times.

$$\frac{ds}{dt} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}. \tag{4}$$

This expression is a complicated function of the parameters and is only equal to $\frac{S}{2\pi}$ for circles and for the infinite series [27].

The geometric properties of each of the component ellipses can be derived from the raw elements of each ellipse matrix. Each ellipse can be described by four geometric properties: semi-major axis length, semi-minor axis length, rotation, and phase shift. The rotation is the angle from the x axis to the major axis of the ellipse, which is defined from $-\pi/2$ to $\pi/2$. The phase shift is the difference in phase from the major axis to the position of $t = 0$ (the ellipse starting position), which is defined from $-\pi$ to π.

These ellipse properties can be derived as follows. First, consider the general form for an ellipse, which is the product of the raw ellipse matrix and the trigonometric basis function vector:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix}. \tag{5}$$

In order to determine the ellipse parameters, consider the matrix for an ellipse with its major axis aligned with the x axis and with no phase shift:

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \tag{6}$$

where A and B are the major and minor semi-axis lengths, respectively. The phasor moves counterclockwise for B positive and clockwise for B negative. The ellipse can be rotated simply by premultiplying the ellipse matrix by a rotation matrix. A phase shift of the ellipse by ϕ_0 means replacing t by $t + \phi_0$. This is the same as a premultiplication of the basis function vector by a rotation matrix or, equivalently, a

postmultiplication of the ellipse matrix. Thus, a rotation of this ellipse by θ and shift by ϕ can be written as a premultiplication and a postmultiplication by rotation matrices:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}. \quad (7)$$

This represents a general ellipse and is thus equivalent to the raw ellipse matrix in (5). Therefore, to find the ellipse parameters given the values of these matrix elements, solve the following four equations that come from identifying corresponding matrix elements for A, B, θ, and ϕ.

$$\begin{aligned} a &= +A\cos\theta\cos\phi - B\sin\theta\sin\phi \\ b &= -A\cos\theta\sin\phi - B\sin\theta\cos\phi \\ c &= +A\sin\theta\cos\phi + B\cos\theta\sin\phi \\ d &= -A\sin\theta\sin\phi + B\cos\theta\cos\phi. \end{aligned} \quad (8)$$

This results in

$$\begin{aligned} A^2 &= \frac{\alpha + \sqrt{\alpha^2 - 4\beta^2}}{2} \\ B^2 &= \frac{2\beta^2}{\alpha + \sqrt{\alpha^2 - 4\beta^2}} \\ \theta &= \tan^{-1}\frac{Ac + Bb}{Aa - Bd} \\ \phi &= \tan^{-1}\frac{Ba - Ad}{Ac + Bb} \end{aligned} \quad (9)$$

where

$$\alpha = a^2 + b^2 + c^2 + d^2, \qquad \beta = ad - bc.$$

By taking A to be positive and B to agree in sign with β, we get a consistent sign convention. These parameters $\boldsymbol{p}_{\text{ref}} = (a_0, c_0, A_1, B_1, \theta_1, \phi_1, \ldots)$ represent the shape in terms of the ellipse properties and will be referred to as the refined parameters.

A further conversion can improve this set by converting the rotation and shift parameters from absolute quantities to values relative to the preceding harmonic and by normalizing the lengths of the axes. This conversion to relative quantities will allow the isolation of an overall rotation parameter and the removal of the overall phase shift ϕ_1, which is arbitrary. Normalizing the lengths of the axes creates an overall scale parameter.

To convert between relative (θ') and absolute (θ) rotations, use

$$\theta'_k = \theta_k - \theta_{k-1} \qquad \theta_k = \sum_{l=1}^{k} \theta'_l. \quad (10)$$

One parameter θ'_1 now determines the overall rotation of the object, and each rotation value θ'_k represents the rotation relative to the preceding harmonic. The conversion of the phase shifts to relative values is done in two parts. The overall phase shift is removed, and then, the relative values are calculated. A phase shift of ϕ_0 applied to the boundary is equivalent to replacing t by $t + \phi_0$. For the kth ellipse, this results in a shift of $k\phi_0$. The overall phase shift of the object can thus be removed by subtracting its effects from the other harmonics and then setting it to zero, that is

$$\phi^*_k = \phi_k - k\phi_1 \qquad \phi^*_1 = 0. \quad (11)$$

The parameters ϕ^*_k represent the absolute phase shifts with the overall phase set to zero. These values can now be converted to and from relative values in the same way as for the rotations:

$$\phi'_k = \phi^*_k - \phi^*_{k-1} \qquad \phi^*_k = \sum_{l=1}^{k} \phi'_l. \quad (12)$$

The parameters ϕ'_k represent the relative phase shifts. The lengths of the axes can be normalized to the first major axis in order to isolate a single parameter that determines the overall scale:

$$A'_k = \frac{A_k}{A_1} \quad \text{for} \quad k \neq 1 \qquad B'_k = \frac{B_k}{A_1}. \quad (13)$$

These parameters $\boldsymbol{p}_{\text{rel}} = (a_0, c_0, A'_1, B'_1, \theta'_1, A'_2, B'_2, \theta'_2, \phi'_2, \ldots)$ express the boundary in terms of the relative ellipse properties. The further conversion to relative parameters shown in (10)–(13) is both continuous and unique, except that the starting point ambiguity has been removed. We have explicit equations for the conversion between the raw coefficients of the Fourier expansion and the refined and relative ellipse parameters.

E. Open Curves

The elliptical Fourier descriptors can also be used for open curves. Open curves are useful for representing objects or parts of objects that do not have a complete boundary, such as organs with openings or blood vessels, and are best described as curved line segments. The curve can be represented as before by two functions $x(t)$ and $y(t)$, but since the curve is open, a straightforward representation of the curve would result in a discontinuity. Analogously to Persoon and Fu [27], this discontinuity can be avoided by having the parameter t start at one end of the line, trace along the contour to the other end, and then retrace the curve in the opposite direction to create a closed path

$$x(t) = x(2\pi - t), \qquad y(t) = y(2\pi - t). \quad (14)$$

This results in functions $x(t)$ and $y(t)$ that are even, and thus, their Fourier sine terms b_k and d_k are zero. The converse, namely, that any elliptic Fourier expansion with b_k and d_k equal to zero for all k results in an even function and thus describes an open curve, is also true. We can thus represent an arbitrary even function in terms of a sinusoidal basis by restricting the basis functions to include only even ones.

$$\phi\text{even} = \left\{ \frac{1}{2\pi}, \frac{\cos x}{\pi}, \frac{\cos 2x}{\pi}, \frac{\cos 3x}{\pi}, \ldots \right\}. \quad (15)$$

This representation can be thought of as decomposing the curve into degenerate ellipses (flattened down to two coincident lines). The equations for the corresponding ellipse parameters are then just

$$A^2 = a^2 + c^2 \qquad B^2 = 0 \qquad \theta = \tan^{-1}\frac{c}{a} \qquad \phi = 0. \quad (16)$$

The ellipses are all degenerate with a fixed starting point at one end, thus forcing both the minor semi-axis length B and the starting point ϕ to be zero. The relative transformations for θ_k and A_k of the previous section can also be applied.

F. Number of Harmonics

The summation in (3) must, in practice, be truncated. This truncation limits the number of parameters and smooths the curve but decreases the accuracy of the representation. Limiting the number of harmonics limits the representation to smooth functions and thus constrains the boundary to be smooth by excluding functions with higher frequency variation. This is analogous to regularization for ill-posed problems [29]. In regularization, a functional is devised that incorporates a smoothness constraint. Here, the solution space is directly restricted to allow only smooth solutions. Smoothing by reconstructing a truncated elliptic Fourier representation is, in general, a good method for smoothing curves that eliminates the problem of shrinkage [22] caused by direct filtering methods. The choice of number of harmonics is a tradeoff between desired accuracy, conciseness, and degree of smoothing. Many biological forms are relatively smooth and unconvoluted and, thus, are well represented by a small number of components. However, if enough parameters are kept, objects with high-frequency contour features may be represented. Unfortunately, there will be a corresponding increase in computational cost.

Giardina and Kuhl [13] derived a bound on the error in representing a contour that is inversely proportional to the number of harmonics and proportional to the integral of the absolute value of the second derivative. Empirically, this bound was found to be too conservative on a variety of shapes by at least a factor of two [19]. Instead, the appropriate number of harmonics to use for the representation can be determined directly because of the prior information about the shape. As will be discussed in the next section, a sample set of boundaries must be obtained. Each boundary is reconstructed varying the number of harmonics, and the error from the boundary is measured. From this procedure, the number of harmonics necessary to reconstruct these curves within a fixed error bound is determined. For most of the examples considered, between four and six terms of the expansion have been used.

G. The Parameter Probability Distributions

The probability distributions associated with the parameters are intended to bias the model toward a particular range of shapes. This prior knowledge comes from experience with a sample of images of the object being delineated when such a sample is available. When prior information is not available, uniform distributions are used for the prior probabilities of the parameters. It will be necessary, however, to supply an initial estimate of the boundary for the optimization process. The images in a sample will differ due to variability in the object shape and the view of the object. Consistency of object shape is often a reliable assumption and leads to peaked distributions for the governing parameters. Consistency of view is not always possible, and thus, the view parameters may have to be determined by some other means. The prior probability distributions can then be estimated from the shapes determined from the sample by decomposing the boundaries into their model parameters and collecting statistics. In order to calculate these statistics, the boundaries of the objects must be determined either by manual segmentation, or alternatively, this method can be run on a set of exemplar images with manual initialization and uniform distributions.

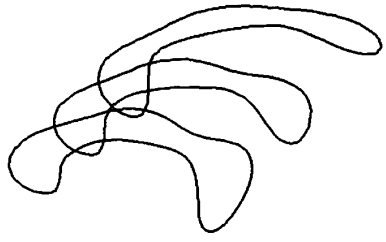

Fig. 2. Example mean curve, shown with curves corresponding to parameters plus and minus one standard deviation.

If a particular distribution is known to govern the parameters, it can be used as the prior probability, although if it is not unimodal, it will make the optimization difficult. Otherwise, if mean and variance information is known, an independent, multivariate Gaussian can be used for the N parameters:

$$\Pr(\boldsymbol{p}) = \prod_{i=1}^{N} \Pr(p_i) = \prod_{i=1}^{N} \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(p_i - m_i)^2}{2\sigma_i^2}}. \quad (17)$$

Here, m_i is the mean of p_i, and σ_i^2 is the variance. An example distribution is shown in Fig. 2. The curve corresponding to the mean parameter values is the middle curve shown in Fig. 2. The curves corresponding to the mean parameter values plus and minus one standard deviation are above and below it.

The Gaussian is the natural form for a probability density, and its use may be understood in terms of information theory. Among probability densities with a given variance, the Gaussian is the one with the maximum entropy [32]. Thus, the Gaussian density follows directly from knowing no information other than a mean and a variance. In general, the parameters in the model may not be independent. Although this effect is not considered in this work, it could easily be incorporated by accumulating the appropriate statistics to construct the covariance matrix.

IV. Boundary Finding Objective Function

In this work, the model is fit to the image by optimization in the parameter space. The feature of interest is shape, and thus, the boundary of the object is the goal and focus of the analysis. The object is expected to be distinguished from the background by some measure of boundary strength (and direction, when available) computed from the image. Only the boundary of the shape is modeled. The boundary value will be modeled as constant along the entire boundary. Focusing on the boundary has the advantage of reducing the computation of the comparison because it is needed only along the boundary. Using a boundary measure has the advantage of improving

the contrast between a good fit and a close fit because slightly misfit areas fit much better than slightly misfit boundaries.

In this section, we derive an objective function that is a measure of fit based on a maximum *a posteriori* formulation. Next, an expression for the likelihood is derived from a simple image noise model. The objective is then simplified for the parametric boundary template. Then, we discuss two issues related to the derivation of the likelihood expression: the independence assumption and the noise. Finally, the boundary measures used are discussed.

A. Bayes Rule

In order to apply the prior knowledge of shape to the problem of boundary determination, we can formulate the problem using a maximum *a posteriori* (or minimum error) criterion using Bayes rule. Consider the problem of boundary determination as one in which the data is an image $b(x,y)$, which could be depicting any one of a set of objects, and $t_{\boldsymbol{p}}(x,y)$ is an image template corresponding to a particular value of the parameter vector $\boldsymbol{p}$. The goal is to determine which object is depicted. In order to determine this, we should find the most probable object based on both the prior information and the image information. In terms of probabilities, if we want to decide to which template $t_{\boldsymbol{p}}$ and image b correspond, we need to evaluate the probability of the template, given the image $\Pr(t_{\boldsymbol{p}}|b)$, and find the maximum over $\boldsymbol{p}$. This can be expressed using Bayes rule, where

$$\Pr(t_{\text{map}}|b) = \max_{\boldsymbol{p}} \Pr(t_{\boldsymbol{p}}|b) = \max_{\boldsymbol{p}} \frac{\Pr(b|t_{\boldsymbol{p}})\Pr(t_{\boldsymbol{p}})}{\Pr(b)}. \quad (18)$$

Here, t_{map} is the maximum *a posteriori* solution, $\Pr(t_{\boldsymbol{p}})$ is the prior probability of template $t_{\boldsymbol{p}}$, and $\Pr(b|t_{\boldsymbol{p}})$ is the conditional probability, or likelihood, of the image given the template. This expression can be simplified by taking the logarithm and eliminating $\Pr(b)$, which is the prior probability of the image data that will be equal for all $\boldsymbol{p}$. Thus, it suffices to maximize

$$\begin{aligned} M(b, t_{\text{map}}) &= \max_{\boldsymbol{p}} M(b, t_{\boldsymbol{p}}) \\ &= \max_{\boldsymbol{p}} \left[\ln \Pr(t_{\boldsymbol{p}}) + \ln \Pr(b|t_{\boldsymbol{p}})\right]. \end{aligned} \quad (19)$$

The function M is the general form of the objective function that will be optimized to find the maximum *a posteriori* solution. This basic form shows the tradeoff or compromise that will be made between prior information $\Pr(t_{\boldsymbol{p}})$ and image-derived information $\Pr(b|t_{\boldsymbol{p}})$. For a uniform prior, this formulation reduces to the maximum likelihood solution. The likelihood $\Pr(b|t_{\boldsymbol{p}})$ can be derived from the image, as described below.

B. Likelihood Derivation

The likelihood of obtaining b given that a particular template $t_{\boldsymbol{p}}$ is present is $\Pr(b|t_{\boldsymbol{p}})$. Consider the image b to be a noise-corrupted version of one of these templates with noise that is independent and additive: $b = t_{\boldsymbol{p}} + n$. This assumption is discussed in Section IV-D. Then, $\Pr(b|t_{\boldsymbol{p}})$ is equivalent to $\Pr(b = t_{\boldsymbol{p}} + n)$ or $\Pr(n = b - t_{\boldsymbol{p}})$. The noise at each pixel $n(x,y)$ equals $b(x,y) - t_{\boldsymbol{p}}(x,y)$ and is governed by the probability density $\Pr(n)$. These events are independent for each point; therefore, the probability for the noise over the entire area $\mathcal{A}$ is just the product of the individual probabilities. Thus, the conditional probability of obtaining b given that it arises from $t_{\boldsymbol{p}}$ is the product of the noise probabilities at each pixel, that is

$$\Pr(b|t_{\boldsymbol{p}}) = \prod_{\mathcal{A}} \Pr(n(x,y)). \quad (20)$$

We make the further assumption (see Section IV-E) that the noise is Gaussian with zero mean and standard deviation σ_n. This gives

$$\Pr(b|t_{\boldsymbol{p}}) = \prod_{\mathcal{A}} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{\left(b(x,y) - t_{\boldsymbol{p}}(x,y)\right)^2}{2\sigma_n^2}}. \quad (21)$$

Now, by taking the logarithm and substituting this result into (19), we can expand the objective function equation to get

$$\begin{aligned} M(b, t_{\boldsymbol{p}}) = \ln \Pr(t_{\boldsymbol{p}}) &+ \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} \\ &- \sum_{\mathcal{A}} \frac{\left(b(x,y) - t_{\boldsymbol{p}}(x,y)\right)^2}{2\sigma_n^2}. \end{aligned} \quad (22)$$

This equation is the maximum *a posteriori* function for images with the assumption of independent Gaussian noise at each pixel. The first term is the logarithm of the prior probability. The second term is a constant. The third term is a sum of squared differences calculation. Because $\sum_{\mathcal{A}} b^2(x,y)$ is a constant, this is approximately equivalent to a correlation if $\sum_{\mathcal{A}} t_{\boldsymbol{p}}^2(x,y)$ does not vary much. A similar interpretation for correlation as likelihood was given by Rosenfeld and Kak [31] and others.

C. Boundary Formulation

The object template $t_{\boldsymbol{p}}(x,y)$ represents the boundary of the object. The templates form a continuum, each having a corresponding value of a parameter vector $\boldsymbol{p}$. The ideal boundary is 1-D, but it can be embedded into a 2-D image. This is done by making $t_{\boldsymbol{p}}(x,y)$ constant along the boundary of the object it represents and zero everywhere else. In order to fit this template to the image, consider $b(x,y)$ to be a boundary measure applied to the raw image data $b(x,y) = b(i(x,y))$. Thus, both $t_{\boldsymbol{p}}$ and b are 2-D functions that represent boundaries. They are summed (or integrated), however, only along a contour.

Because the template has support only along the boundary, it is not necessary to sum over the entire image area for terms involving the template but only over the curve represented by the template. Equation (22) can be rewritten

$$\begin{aligned} M(b, t_{\boldsymbol{p}}) = \ln \Pr(t_{\boldsymbol{p}}) &+ \sum_{\mathcal{A}} \ln \frac{1}{\sqrt{2\pi}\sigma_n} - \frac{1}{2\sigma_n^2} \\ &\cdot \left(\sum_{\mathcal{A}} b^2(x,y) + \sum_{C_{\boldsymbol{p}}} \left(-2b(x,y)t_{\boldsymbol{p}}(x,y) + t_{\boldsymbol{p}}^2(x,y) \right) \right) \end{aligned} \quad (23)$$

where $\mathcal{C}_{\boldsymbol{p}}$ is the curve defined by the boundary $(x(\boldsymbol{p}), y(\boldsymbol{p}))$ in template $t_{\boldsymbol{p}}$. This can be simplified because $t_{\boldsymbol{p}}(x, y)$ is constant over the curve that it defines:

$$M(b, t_{\boldsymbol{p}}) = \ln \Pr(t_i) + \sum_{\mathcal{A}} \left(\ln \frac{1}{\sqrt{2\pi}\sigma_n} - \frac{b^2(x,y)}{2\sigma_n^2} \right) + \frac{1}{2\sigma_n^2} \sum_{\mathcal{C}_{\boldsymbol{p}}} \left(2b(x,y)k - k^2\right) \quad (24)$$

where k is the magnitude of the template at any point taken to be constant and chosen to be the maximum boundary response. The function M can be simplified further by removing the terms that do not depend on the different possible templates since we want the maximum over $\boldsymbol{p}$. We can also remove the k^2 term because it is a function only of the length of the curve, which we can approximate as a constant because it will not vary significantly.

$$M(b, t_{\boldsymbol{p}}) = \ln \Pr(t_{\boldsymbol{p}}) + \frac{1}{\sigma_n^2} \sum_{\mathcal{C}_{\boldsymbol{p}}} b(x,y)k. \quad (25)$$

This equation is the maximum *a posteriori* objective for boundary templates. The first term is the bias due to the prior probability. The second term is simply a correlation of a boundary template with the boundary strength in the image and is thus a kind of matched filter [31].

We can also consider the boundary to be a vector-valued quantity in which the magnitude of the vector represents the strength of the boundary, and the direction of the vector is the direction of the tangent to the boundary. This means that $t_i(x, y)$ has a constant magnitude along the boundary of the object it represents and a direction equal to the tangent to the boundary. The corresponding $\boldsymbol{k}$ is now a function of position along the curve:

$$\boldsymbol{k}(x, y) = k \begin{bmatrix} \frac{\partial x(\boldsymbol{p},s)}{\partial s} \\ \frac{\partial y(\boldsymbol{p},s)}{\partial s} \end{bmatrix}. \quad (26)$$

The boundary measure $\boldsymbol{b}$ is a measure of both boundary magnitude and direction. Equation (25) can be interpreted as vector valued and rewritten using the dot product.

The continuous version of (25) is

$$M(b, \boldsymbol{p}) = \ln \Pr(\boldsymbol{p}) + \frac{k}{\sigma_n^2} \int_{\mathcal{C}_{\boldsymbol{p}}} b(x,y)ds. \quad (27)$$

This line integral can be written as a regular definite integral:

$$M(b, \boldsymbol{p}) = \ln \Pr(\boldsymbol{p}) + \frac{k}{\sigma_n^2} \int_0^S b(x(\boldsymbol{p},s), y(\boldsymbol{p},s))ds. \quad (28)$$

If a uniform prior were used, the first term would be a constant and, thus, inconsequential in the maximization. We can, instead, expand the first term using the Gaussian distribution shown in (17) to get

$$M(b, \boldsymbol{p}) = \sum_{i=1}^{N} \left[\ln \left(\frac{1}{\sigma_i \sqrt{2\pi}} \right) - \frac{(p_i - m_i)^2}{2\sigma_i^2} \right] + \frac{k}{\sigma_n^2} \int_0^S b(x(\boldsymbol{p},s), y(\boldsymbol{p},s))\, ds. \quad (29)$$

In the vector-valued version of this equation, the integrand is $\boldsymbol{b}(x(\boldsymbol{p}, s), y(\boldsymbol{p}, s)) \cdot \boldsymbol{k}$.

Equation (29) is the objective function expressed in terms of the parameters that are used for boundary finding. The first term of this objective function is the contribution of the prior probability of the parameter vector. The influence of this term is controlled by the variance of the prior probability. If the variance of the prior is greater, the influence of this term is smaller. The prior also determines the starting point for the optimization process. The second term is the contribution of the image information.

D. Independence Assumption

The boundary measure at each pixel is calculated from a neighborhood of pixels, and therefore, the values are correlated. Since the boundary measure is a known function of the neighboring pixels, this correlation can be determined explicitly. In addition, the original image pixel values are most likely correlated, but the correlation is, in general, unknown. The problem with including the correlation is that it excessively increases the complexity of the problem. In order to account for the correlation between boundary measure values, a covariance matrix for all of the boundary values would have to be constructed and inverted. To avoid this complication, we invoke an assumption of independence. Cooper [9] made the same independence assumption for a maximum likelihood approach to boundary finding. He compared the probability of error with and without the independence assumption for some simple examples and found that the assumption did not alter the performance significantly. The independence assumption amounts to ignoring information (not assuming additional information) by the principle of maximum entropy [32].

E. Noise

The noise, as described by the formulation in (21), should be thought of as not just the degradation of the signal due to the imaging process but also as the combined effect of many factors such as other objects, occlusion, and boundary measurement. These factors are, in general, impossible to model explicitly. Instead, we assume that it can be described by a Gaussian density with zero mean. A nonzero mean would only change the expression by a constant. For the variance of the noise σ_n^2, we could estimate it as we did the model parameters, that is, measure it on solutions obtained either manually or with this method using a uniform prior. Although the performance depends on σ_n^2, it is not too sensitive to it. As can be seen in (25), σ_n^2 represents the weight of the image information and, thus, the relative importance of the prior information and the likelihood. If the noise is greater, the prior is more important.

F. Boundary Measures

Any measure that indicates a change in some property that distinguishes the object from the background could be used as a boundary measure. A natural candidate for many images is the gray-level gradient. The gray-level gradient can be calculated by first smoothing with a Gaussian to reduce the

effect of noise and simplify the image followed by a finite difference approximation to the partial derivatives in order to control smoothing independently. A good local approximation to the gradient that has components that are unbiased for the same point was described by Horn [16]. It gives the average of two finite difference approximations for a point midway between pixels. The perpendicular to the gradient is used for comparison with the tangent of the curve.

Measures that respond to line strength are useful when the object is delineated by a border of a different gray level. The gray level itself works as a line indicator when it is relatively high at the boundary. A more general line detector is the Laplacian. It works as a line detector by acting as a nondirectional template for a line in that it has a low center and a high surround. The gray-level Laplacian is calculated after smoothing using a 3×3 discrete approximation [16].

V. Boundary Parameter Optimization

The problem to be solved is that of maximizing the objective function $M(\boldsymbol{p})$. The objective function we are solving is not, in general, convex but depends ultimately on the gray-level surface shape of the image. However, the prior probability term in the objective function is quadratic, and it dominates on the tails of the distributions, making distant points in the space nonoptimal. The starting point for the optimization will be taken to be the maximum of the prior distributions. The global optimum probably will be near the starting point, and thus, a local optimum is likely to be a global optimum. The degree to which this is true depends on the width of the distributions. Since a local optimization method is likely to be sufficient, although there is still the possibility of converging to a poor local maximum, the excessive computation involved in finding a global optimum is deemed not necessary. Poor convergence can be identified by a corresponding low objective function value and verified visually. Smoothing can also be used to avoid getting trapped in a local maximum (see Section V-C).

Most numerical optimization methods require the gradient of the objective function because it gives the direction of greatest increase in the function value [28]. The gradient therefore provides the best direction to move in the space in order to maximize the objective function. An analytic form for the gradient must be derived in order to make use of these methods. A discrete approximation is sensitive to errors due to poor choice of step size and is computationally intensive. Methods that do not require gradient information have the advantage of allowing greater flexibility in formulating the objective function and in choosing the parameter space in that they free us of the restriction of differentiability. Two optimization methods were investigated for the implementation: Powell's direction set method [30], which does not use gradient information, and continuous gradient ascent [28].

A. Powell's Direction Set Method

Powell's method chooses a set of directions in the parameter space to move in and then alters the directions based on the progress made. For each direction, a line maximization is performed by standard means such as golden-section search. These methods require a known interval in which the maximum lies. One end of the interval is $\lambda = 0$. The other end of the interval can be found by taking larger and larger steps in the direction that the function increases until the function decreases again. Unfortunately, there is no basis for choosing the size of the steps taken in order to guarantee remaining in the neighborhood of one local maximum.

For this method, we use the relative parameters $\boldsymbol{p}_{\text{rel}}$ because we are not concerned with the complexity of the gradient. In spite of the difficulties of line maximizations, this method works well, but the number of function evaluations is large. This optimization method was applied to a range of problems including those shown in Section VI. On average, execution took 30 min of CPU time on a Vaxstation II.

B. Continuous Gradient Ascent

Powell's method is slow to converge because of the lack of gradient information and the use of line maximizations. Most gradient methods, such as steepest ascent or conjugate gradient methods, also use line maximization. The rationale for line maximization is to make the most out of each gradient computation because of the computational cost. However, the gradient direction is only the steepest direction in the small neighborhood about the point of evaluation. The cost of the many function evaluations required for line maximization can outweigh the cost of additional gradient evaluations. In addition, there is still the problem of obtaining a local optimum using line maximization with a multimodal objective.

An alternative approach that avoids the use of line maximizations is continuous gradient ascent [28], which takes small steps in the direction of the gradient. The method always moves in the direction of greatest increase in order to save many function evaluations. If the gradient computation is comparable with the function computation, the overall computation can be greatly reduced.

The rate of convergence of continuous gradient descent has been shown to be comparable with, and potentially much better than, steepest descent [33]. The gradient calculation is described in Appendix A. We will use only $\boldsymbol{p}_{\text{raw}}$ and $\boldsymbol{p}_{\text{ref}}$ and not $\boldsymbol{p}_{\text{rel}}$ because of the complexity of the gradient with respect to the relative parameters. For a range of problems including those in Section VI, this optimization method took an average of 2 min of CPU time on a Vaxstation II.

C. Smoothing

In order to further reduce the chance that the optimization process will converge to a poor local maximum, smoothing is used. The amount of smoothing necessary for an image depends on the amount of noise, which is not usually known. A certain amount of smoothing is necessary to lessen the effect of noise; further smoothing simplifies the image. Scale-space, or continuation, methods solve multimodal optimization problems by first optimizing at a coarse scale (high smoothing) and then tracking the optimal point to increasingly finer scales (less smoothing) [35]. The Gaussian is used as the smoothing filter. The Gaussian is the optimally localized function that is smooth in both space and frequency. For the

current problem, when the image is smoothed, the objective function is smoothed, and the optimization space is simplified. A smoother and simpler structure in the optimization space makes the optimum easier to find because the local region of unimodality grows larger. However, too much smoothing on complex scenes is a mistake once the smoothing goes beyond simplifying individual objects. In order to avoid the problems of too much smoothing while still simplifying the image somewhat, we have adopted a simple two-level approach. The optimization is first done using an intermediate amount of smoothing (σ ranging from 2.5 to 5.0). The result of that optimization is then used as the initial estimate for a second optimization, which is done on a slightly smoothed version ($\sigma = 1.0$), in order to accurately localize the boundary.

VI. Experiments

The boundary finding system was evaluated by testing on real and synthetic images. In particular, the dependence of its accuracy on prior information and image quality was investigated.

A. Evaluation

For synthetic images, the true boundary is known, and thus, a quantitative measure of error can be devised. For real images, we must rely on qualitative evaluation. A natural quantitative measure of error is one that is equal to zero for a perfect match and gets larger as the boundaries become further apart in terms of Euclidean distance. Quantitative evaluation is a relatively rare and often neglected component of computer vision. Edge detector performance, which is a somewhat different problem, has been addressed by Abdou and Pratt [1]. The measure they developed was based on the distance between the detected edge and the actual edge but was limited to straight edges. Cooper [9] used a similar measure that was limited to perturbations of straight lines. All of these approaches simplify the problem by reducing it to 1-D, although the problem is inherently 2-D.

In order to evaluate the error while retaining the two dimensionality of the problem, we need to establish a correspondence between the points on the two curves. The appropriate error measure is then the average distance between the corresponding points on the two curves. The correspondence can be determined by finding the offset between the curves that produces the minimum average error. In discrete form, the curves must first be made commensurable by resampling them to an equal number of equally spaced points using simple linear interpolation. The error ϵ between the two curves is then defined by

$$\epsilon(v, v') = \min_{0 \le t_0 \le n} \frac{1}{n} \sum_{t=0}^{n} \|v(t) - v'(t + t_0)\| \qquad (30)$$

where v is the true curve, v' is the measured boundary, and t_0 is the offset. The average error has to be evaluated at all possible offsets in order to determine the minimum. The computational burden is not important, however, since this is only done for evaluation purposes and not in actual use.

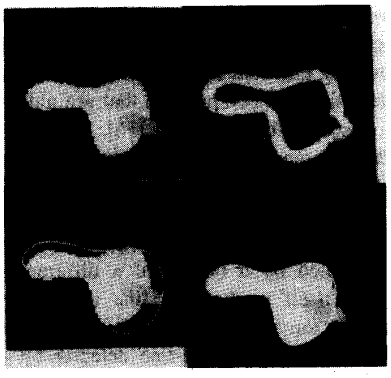

Fig. 3. Synthetic image example. Top left: synthetic image (96 × 96). Top right: gray-level gradient magnitude ($\sigma = 3.0$). Bottom left: initial contour (four harmonics). Bottom right: final contour on target shape.

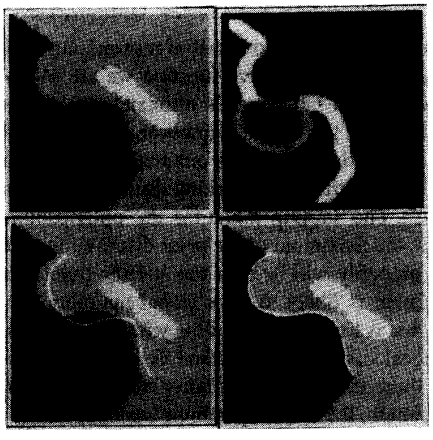

Fig. 4. Synthetic image open curve example. Top left: synthetic image (64 × 64). Top right: gray-level gradient magnitude ($\sigma = 3.0$). Bottom left: initial contour (12 harmonics). Bottom right: final contour on target shape.

B. Synthetic

The image shown in Fig. 3 is a simple synthetic Mondrian image where the target object (the brightest) is partially occluded by one object on the right and overlaps another. The extraneous objects are potential sources of choice or confusion because alternate boundaries are plausible, especially at the right side of the target and at the center of the top boundary. The probability distributions for the parameter vector $\mathbf{p}$ were derived from a set of manually traced contours. The initial curve superimposed is defined by the mean parameter vector. The starting boundary only roughly agrees with the target in terms of shape and location. The final curve accurately delineates the target with approximately one half pixel average error. The curve correctly avoids both the overlapped and the occluding object both because of the limit on the number of harmonics (four) and the bias due to the probability distributions on the parameters.

Another synthetic image is shown in Fig. 4. The target is the s-shaped portion of the boundary of the large object. This structure is described by an open curve model. The other objects in the image are potential sources of confusion, as in

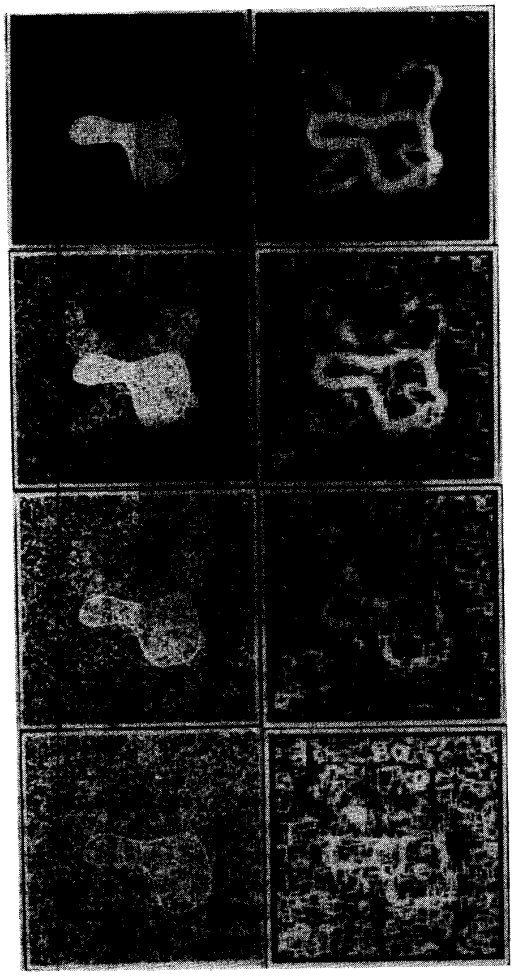

Fig. 5. Noise experiment images. Left, top to bottom: image from Fig. 3 with Gaussian noise added with SNR of 5.0, 2.5, 1.0, and 0.5. Each shown with final contour. Right, top to bottom: corresponding gray-level gradient magnitude ($\sigma = 3.0$).

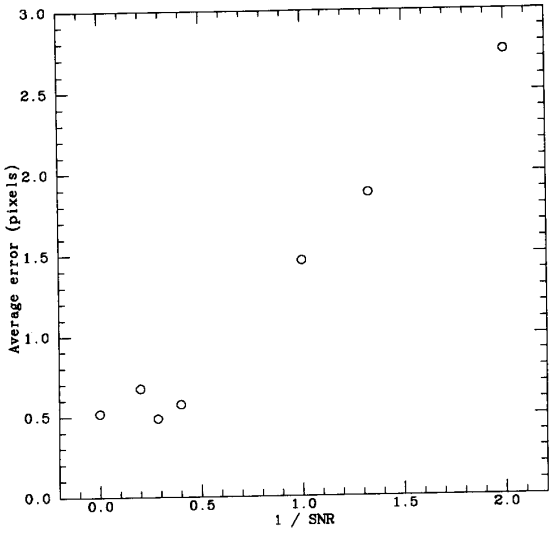

Fig. 6. Sensitivity to noise experiment.

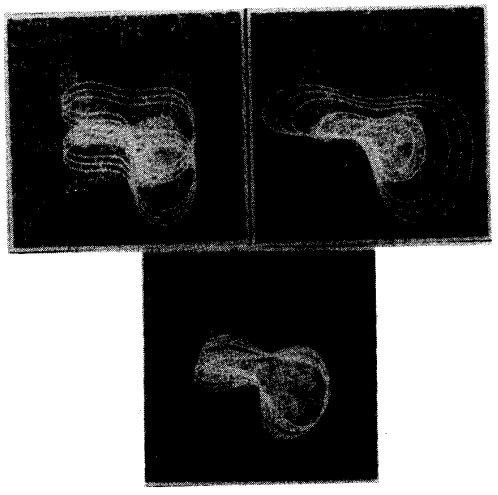

Fig. 7. Parameter sensitivity experiment parameters. Top left: image from Fig. 3 with Gaussian noise added (SNR = 2.5) shown with initial curves for range of vertical translation tested. Top right: Image with initial curves for range of scale tested. Bottom: image with initial curves for range of rotation tested.

the previous example. The final curve accurately describes the target and avoids the other objects.

C. Varying Noise

The effect of noise on the performance of the system can be investigated by adding different amounts of noise to a synthetic image and comparing the resulting boundaries to the true boundary. The synthetic image shown in Fig. 3 was altered by the addition of Gaussian noise of zero mean and varying standard deviation. Examples of the noisy images are shown in Fig. 5. The accuracy of the results of running the optimization on the noisy images, using the same distributions for each, is shown in Fig. 6. Signal-to-noise ratio (SNR) is defined here as the ratio of gray-level contrast between object and background to the standard deviation of Gaussian noise. The error shown is from the boundary from which the image was constructed. The accuracy of the resulting contours is good (~ 0.5 pixels) for SNR > 1 and steadily worsens as the noise increases. Most of the error seems to be due to confusion with the overlapping object on the right.

D. Varying Initial Parameters

The effect of the initial values of the parameters on the performance can be investigated by examining the results of running the same problem from different starting points. Here, we used the same synthetic image with SNR = 2.5 constructed as above. The parameters varied were vertical translation, scale, and rotation. The other parameters were held constant, whereas each of the above three parameters were varied individually. The range of initial curves for the parameters tested are shown in Fig. 7 with the image used. This image,

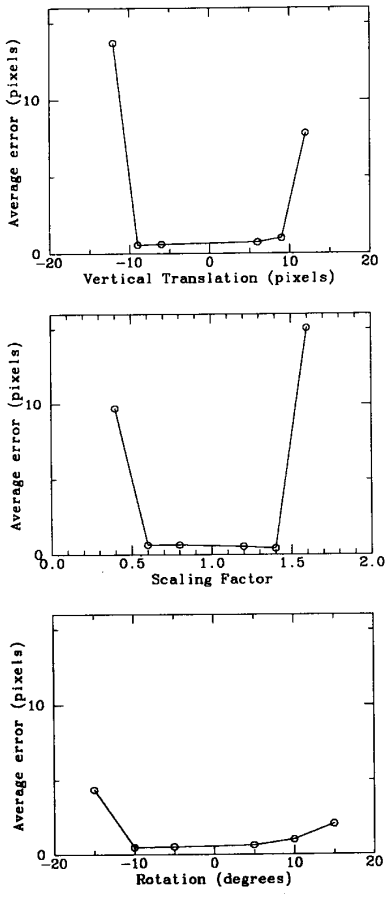

Fig. 8. Sensitivity to initial parameters experiment.

although it is synthetic, has a reasonable amount of noise and complication. The accuracy of the resulting optimized boundaries using these different starting points is shown in Fig. 8. Each parameter has a range within which the solution can be found to be reliable. Once the parameters are varied beyond that range, the result will converge to false local minima corresponding to nearby features. This region of success or capture about the true boundary depends on the quality of the image, the degree of smoothing, and the particular problem. False minima can be distinguished, however, both visually and by the relative value of the objective function.

E. Varying Bias

This experiment is designed to show different results from two different prior probability distributions applied to the same image. To show that the difference is due to the prior probability term in the objective function and not simply the starting point, the distributions will have the same mean value but different variances. A synthetic Mondrian image, which is shown in Fig. 9, was designed containing two similar objects. The light object corresponds to a rotation of the mean prior curve. The dark object underneath it is a scaled version of the mean curve. The prior distribution can be biased toward

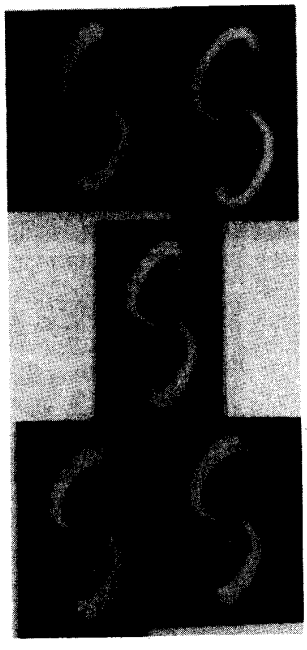

Fig. 9. Bias experiment. Top left: synthetic image (64 × 64). Top right: gray-level gradient magnitude ($\sigma = 1.5$). Middle: initial contour (six harmonics). Bottom left: final contour, biased to scaled target shape. Bottom right: final contour, biased to rotated target shape.

finding the light object by having a wide distribution on the rotation parameter and narrow distributions on the others. Conversely, a wide distribution on the scale parameter and narrow distributions on the others will bias the optimization to the dark object. These two different prior distributions were applied to the image, and the results are shown in Fig. 9.

F. Real Images

The deformable object boundary finding method has been applied to a variety of objects from real images with an emphasis on heart and brain images using primarily magnetic resonance images. The results of the method applied to the problem of delineating the corpus callosum in the human brain from magnetic resonance images are shown in Fig. 10. In these images, the corpus callosum is separated from the rest of the brain by a dark line. In this case, we used the positive magnitude of the Laplacian of the Gaussian as a line detector. The final contour succeeds in delineating the structure properly.

Magnetic resonance is becoming increasingly important for cardiac imaging as acquisition rates increase into the range required for imaging the moving heart. In Fig. 11, a transaxial cardiac image shows a section through the left ventricular wall. Here, the endocardial (inner) and epicardial (outer) walls of the left ventricle are objects to be delineated. The results of the two separate optimizations are shown.

In Fig. 12, we show a simple but effective approach to temporal sequence analysis applied to a cardiac motion sequence from magnetic resonance. By delineating an object in successive temporal frames, the motion of its boundary can be

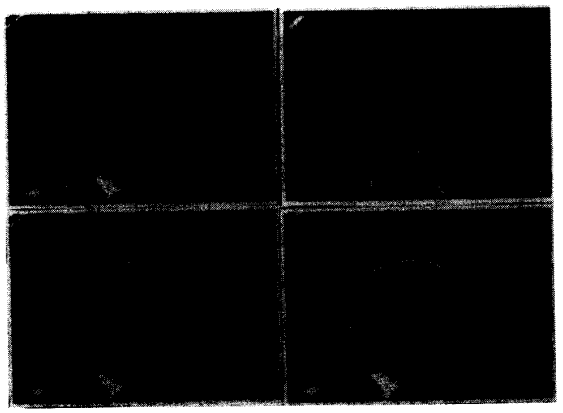

Fig. 10. Magnetic resonance midbrain sagittal image example. Top left: magnetic resonance image (146 × 106). Top right: positive magnitude of the Laplacian of the Gaussian ($\sigma = 2.2$). Bottom left: initial contour (six harmonics). Bottom right: final contour on the corpus callosum of the brain.

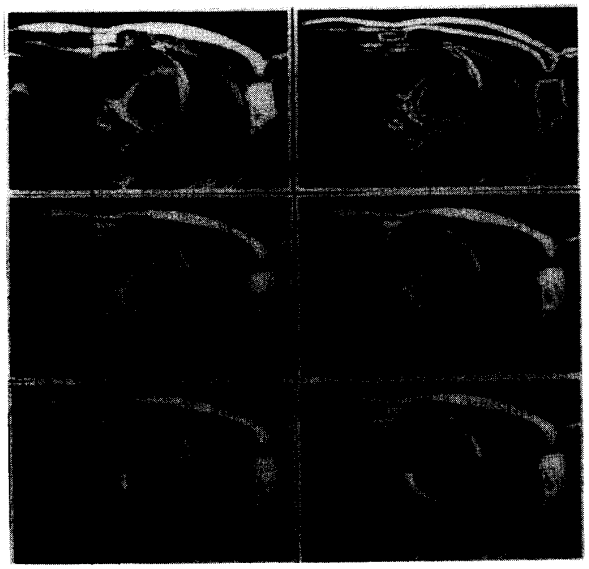

Fig. 11. Magnetic resonance transaxial cardiac image example. Top left: Magnetic resonance image (256 × 156). Top right: gray-level gradient magnitude ($\sigma = 4.0$). Middle left: initial contour on endocardium (four harmonics). Bottom left: final contour on endocardium of the left ventricle. Middle right: initial contour on the epicardium (four harmonics). Bottom right: final contour on the epicardium of the left ventricle.

inferred. The delineation problem is first solved on the initial frame of the sequence. Then, the solution is used as the initial boundary for the next frame since the boundary in each frame should be close to each preceding frame.

VII. Conclusion

This work presents a general boundary finding system for images of simple natural objects. The goal of this work was to incorporate model-based information about global shape into boundary finding for continuously deformable objects by augmenting a shape parametrization with probabilistic

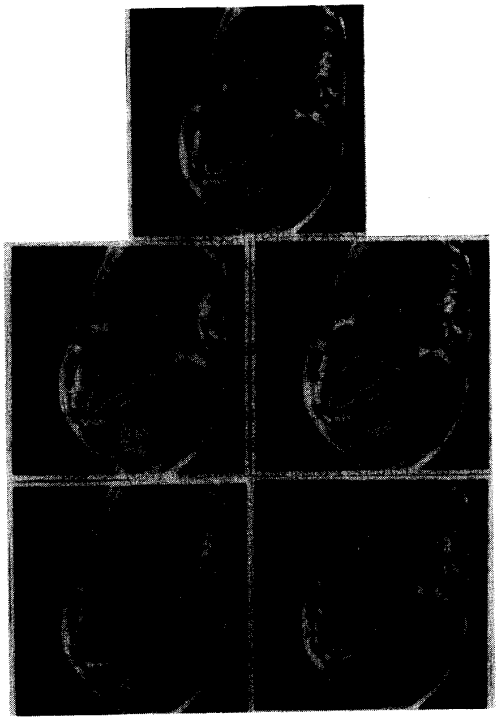

Fig. 12. Magnetic resonance oblique sagittal cardiac image motion example. Top: magnetic resonance image (256 × 256) with initial contour (four harmonics). Middle left: final contour on the endocardium of the left ventricle (frame 1). Middle right: final contour (frame 2). Bottom left: final contour (frame 3). Bottom right: final contour (frame 4).

information. It was found to perform well at delineating structures from testing on real and synthetic images and to be relatively insensitive to the problems of broken boundaries and spurious edges from nearby objects. The flexibility of the model, both in terms of its probabilistic nature and the parametric representation, make this an attractive method for boundary finding.

There are, of course, areas of potential improvement for this work. Further analysis of the correlation between the values of the calculated boundary measure would help to better understand the correlation's effect on the objective function. The calculation of the gradient of the objective function with respect to the relative parameters could be attempted in spite of the complexity. Since one of the main benefits of the relative parameters is the isolation of the view parameters (translation, rotation, and scale), perhaps a simpler transformation that isolates the view parameters could be devised. Because the initial estimates of the view parameters may not be very good, an additional process to determine them could be added. This could involve an initial exhaustive coarse search over just those parameters. If this were done at a low resolution, the computation might not be excessive. Additional information, such as other low-level features or constraints between objects, might also help to guide the initial placement.

Future directions include incorporating constraints and extending the parametrization to curves and surfaces in 3-D.

Initial work in this direction has been described [33]. The framework presented here could also, perhaps, be used with other shape parametrizations better suited to man-made objects with straight sides and corners. The method could also be extended to object recognition where an image is fit to each of the models for different objects in a database. The correct model will result in the best fit because it will be the closest in the parameter space. The boundary finder and some of the ideas from this work have also been applied to the problem of contour-based deformable object motion [11].

Appendix A
Gradient Formulation

A practical gradient calculation may be obtained by using a numerical approximation based directly on the objective function calculation. First, differentiate (28) to get

$$\nabla M(\boldsymbol{p}) = \frac{\partial \ln \Pr(\boldsymbol{p})}{\partial \boldsymbol{p}} - \frac{k}{2\sigma^2}\int_0^S \frac{\partial b(x(\boldsymbol{p},s),y(\boldsymbol{p},s))}{\partial x}\frac{\partial x}{\partial \boldsymbol{p}} + \frac{\partial b(x(\boldsymbol{p},s),y(\boldsymbol{p},s))}{\partial y}\frac{\partial y}{\partial \boldsymbol{p}}ds. \tag{31}$$

In the above equation, we have used the simplifying approximation that ds is not a function of $\boldsymbol{p}$. For Gaussian prior distributions, as in (17), the first term is

$$\frac{\partial \ln \Pr(\boldsymbol{p})}{\partial \boldsymbol{p}} = -\frac{p_i - m_i}{2\sigma_i^2}. \tag{32}$$

The partials of b, which is the boundary measure image, with respect to x and y can be calculated using a central discrete divided difference approximation.

The partials of $x(\boldsymbol{p},s)$ and $y(\boldsymbol{p},s)$ with respect to $\boldsymbol{p}$ can be calculated by noting that

$$\frac{\partial x(\boldsymbol{p},s)}{\partial \boldsymbol{p}} = \frac{\partial x(\boldsymbol{p},t(s))}{\partial \boldsymbol{p}} \tag{33}$$

and that

$$t(s) = t_i + \frac{(t_{i+1}-t_i)(s-s_i)}{s_{i+1}-s_i}. \tag{34}$$

Then, from the Fourier relation in (3), we can calculate the partials of x and y with respect to the raw parameters $\boldsymbol{p}_{\text{raw}}$:

$$\begin{array}{lll} \frac{\partial x(t)}{\partial a_0} = 1 & \frac{\partial x(t)}{\partial a_k} = \cos kt & \frac{\partial x(t)}{\partial b_k} = \sin kt \\ \frac{\partial x(t)}{\partial c_0} = 0 & \frac{\partial x(t)}{\partial c_k} = 0 & \frac{\partial x(t)}{\partial d_k} = 0 \\ \\ \frac{\partial y(t)}{\partial a_0} = 0 & \frac{\partial y(t)}{\partial a_k} = 0 & \frac{\partial y(t)}{\partial b_k} = 0 \\ \frac{\partial y(t)}{\partial c_0} = 1 & \frac{\partial y(t)}{\partial c_k} = \cos kt & \frac{\partial y(t)}{\partial d_k} = \sin kt. \end{array} \tag{35}$$

If we use the refined ellipse parameters $\boldsymbol{p}_{\text{ref}}$ (excluding the relative transformation), we get

$$\begin{array}{l} \frac{\partial x(t)}{\partial a_0} = 1 \\ \frac{\partial x(t)}{\partial c_0} = 0 \\ \frac{\partial x(t)}{\partial A_k} = \cos\theta_k \cos(\phi_k + kt) \\ \frac{\partial x(t)}{\partial B_k} = -\sin\theta_k \sin(\phi_k + kt) \\ \frac{\partial x(t)}{\partial \theta_k} = -A_k \sin\theta_k \cos(\phi_k + kt) - B_k \cos\theta_k \sin(\phi_k + kt) \\ \frac{\partial x(t)}{\partial \phi_k} = -A_k \cos\theta_k \sin(\phi_k + kt) - B_k \sin\theta_k \cos(\phi_k + kt) \\ \\ \frac{\partial y(t)}{\partial a_0} = 0 \\ \frac{\partial y(t)}{\partial c_0} = 1 \\ \frac{\partial y(t)}{\partial A_k} = \sin\theta_k \cos(\phi_k + kt) \\ \frac{\partial y(t)}{\partial B_k} = \cos\theta_k \sin(\phi_k + kt) \\ \frac{\partial y(t)}{\partial \theta_k} = A_k \cos\theta_k \cos(\phi_k + kt) - B_k \sin\theta_k \sin(\phi_k + kt) \\ \frac{\partial y(t)}{\partial \phi_k} = -A_k \sin\theta_k \sin(\phi_k + kt) + B_k \cos\theta_k \cos(\phi_k + kt). \end{array} \tag{36}$$

Differentiation with respect to the relative parameters $\boldsymbol{p}_{\text{rel}}$ will result in a much more complex expression because the interdependencies between the parameters greatly increase.

The vector valued version of the objective function can be differentiated similarly to the scalar one. From (28) and (26), we get

$$\nabla M(\boldsymbol{p}) = \frac{\partial \ln \Pr(\boldsymbol{p})}{\partial \boldsymbol{p}} - \frac{k}{2\sigma^2}\int_0^S \left[\frac{\partial x}{\partial s}\frac{\partial b_x(x,y)}{\partial x} + \frac{\partial y}{\partial s}\frac{\partial b_y(x,y)}{\partial x}\right]\frac{\partial x}{\partial \boldsymbol{p}} + \left[\frac{\partial x}{\partial s}\frac{\partial b_x(x,y)}{\partial y} + \frac{\partial y}{\partial s}\frac{\partial b_y(x,y)}{\partial y}\right]\frac{\partial y}{\partial \boldsymbol{p}}\,ds \tag{37}$$

where b_x and b_y are the x and y components of b, and we have dropped the explicit notation of the dependence of x and y on $\boldsymbol{p}$ and s. Here, we have also approximated by ignoring the terms $\partial/\partial\boldsymbol{p}(\partial x/\partial s)$ and $\partial/\partial\boldsymbol{p}(\partial y/\partial s)$. The above expression requires the calculation of $\partial x/\partial s$ and $\partial y/\partial s$. These, along with the partials of b_x and b_y with respect to x and y, can be calculated using a discrete divided difference approximation. The other terms are calculated as above.

To numerically integrate, we use the x, y points generated by the template-making procedure along with the associated lengths and $t(s)$ values. The lengths are the weight for each term in the integration. The template is relatively computationally expensive to generate, but it only has to be computed once for each gradient calculation. In fact, the function value and

the gradient can be calculated from one template generation. Overall, the gradient computation is comparable with objective computation and practical to use.

REFERENCES

[1] I. E. Abdou and W. K. Pratt. "Quantitative design and evaluation of enhancement/thresholding edge detectors." *Proc. IEEE*, vol. 67, no. 5, pp. 753–763, May 1979.

[2] R. Bajcsy and S. Kovačič. "Multiresolution elastic matching." *Comput. Vision Graphics Image Processing*, vol. 46, pp. 1–21, 1989.

[3] R. Bajcsy and F. Solina, "Three dimensional object representation revisited," in *Proc. Int. Conf. Comput. Vision*, June 1987, pp. 231–239.

[4] D. H. Ballard. "Generalizing the Hough transform to detect arbitrary shapes," *Patt. Recogn.*, vol. 13, no. 2, pp. 111–122, 1981.

[5] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[6] P. J. Besl. "Geometric modeling and computer vision," *Proc. IEEE*, vol. 76, no. 8, pp. 936–958, Aug. 1988.

[7] D. J. Burr, "A dynamic model for image registration," *Comput. Graphics Image Processing*, vol. 15, pp. 102–112, 1981.

[8] Y. Chow, U. Grenander, and D. M. Keenan, "Hands: A pattern theoretic study of biological shapes." Monograph, Div. Applied Math., Brown Univ., Providence, RI, 1989.

[9] D. B. Cooper, "Maximum likelihood estimation of Markov-process blob boundaries in noisy images." *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI–1, no. 4, pp. 372–384, Oct. 1979.

[10] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[11] J. S. Duncan, R. L. Owen, L. H. Staib, and P. Anandan, "Measurement of nonrigid motion in images using contour shape descriptors," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, June 1991, pp. 318–324.

[12] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures." *IEEE Trans. Comput.*, vol. C-22, pp. 67–92, Jan. 1973.

[13] C. R. Giardina and F. P. Kuhl. "Accuracy of curve approximation by harmonically related vectors with elliptical loci," *Comput. Graphics Image Processing*, vol. 6, pp. 277–285, 1977.

[14] G. H. Granlund, "Fourier preprocessing for hand print character recognition," *IEEE Trans. Comput.*, vol. C-21, pp. 195–201, Feb. 1972.

[15] C. W. K. Gritton and E. A. Parrish, Jr. "Boundary location from an initial plan: The bead chain algorithm," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI–5, no. 1, pp. 8–13, Jan. 1983.

[16] B. K. P. Horn. *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[17] B. K. P. Horn and E. J. Weldon. "Filtering closed curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI–8, no. 5, pp. 665–668, Aug. 1986.

[18] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[19] F. P. Kuhl and C. R. Giardina, "Elliptic Fourier features of a closed contour," *Comput. Graphics Image Processing*, vol. 18, pp. 236–258, 1982.

[20] C. Lin and C. Hwang, "New forms of shape invariants from elliptic Fourier descriptors," *Patt. Recogn.*, vol. 20, no. 5, pp. 535–545, 1987.

[21] D. Lowe, "Three dimensional object recognition from single two-dimensional images," *Artificial Intell.*, vol. 31, no. 3, pp. 355–396, 1987.

[22] ——, "Organization of smooth image curves at multiple scales," in *Proc. Int. Conf. Comput. Vision*, Dec. 1988, pp. 558–567.

[23] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Commun. ACM*, vol. 19, no. 2, pp. 73–83, Feb. 1976.

[24] O. R. Mitchell and T. A. Grogan, "Global and partial shape discrimination for computer vision," *Optical Eng.*, vol. 23, no. 5, pp. 484–491, 1984.

[25] R. Mohan and R. Nevatia, "Segmentation and description based on perceptual organization," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, June 1989, pp. 333–34.

[26] U. Montanari, "On the optimal detection of curves in noisy pictures," *Commun. ACM*, vol. 14, no. 5, pp. 335–345, May 1971.

[27] E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI–8, no. 3, pp. 388–397, May 1986.

[28] D. A. Pierre, *Optimization Theory with Applications*. New York: Dover, 1986.

[29] T. Poggio and V. Torre, "Ill-posed problems and regularization analysis in early vision," Memo 773, MIT AI Lab., Apr. 1984.

[30] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.

[31] A. Rosenfeld and A. C. Kak, *Digital Picture Processing* (2nd Ed.). Orlando, FL: Academic, 1982, vol. 2.

[32] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, 1948.

[33] L. H. Staib, "Parametrically deformable contour models for image analysis," Ph.D. thesis, Yale Univ., New Haven, CT, 1990.

[34] B. Widrow, "The "rubber mask" technique–I and II," *Patt. Recogn.*, vol. 5, pp. 175–211, 1973.

[35] A. Witkin, D. Terzopoulos, and M. Kass, "Signal matching through scale space," *Int. J. Comput. Vision*, vol. 1, pp. 133–144, 1987.

[36] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, June 1989, pp. 104–109.

Lawrence H. Staib (S'86–M'91) received the A.B. degree in physics from Cornell University, Ithaca, NY, in 1982. He received the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 1986, 1987, and 1990, respectively.

He held a National Library of Medicine Postdoctoral Fellowship and is currently an Assistant Professor in the Department of Diagnostic Radiology at Yale University. His research interests are in computer vision and medical image analysis.

James S. Duncan (S'72–M'82–SM'92) received the B.S.E.E. degree from Lafayette College, Easton, PA, in 1972, the M.S. degree in engineering from the University of California, Los Angeles, in 1975, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1982.

He joined the staff of Hughes Aircraft Company, Electro- Optical and Data Systems Group, in 1973 and participated in research and development projects related to signal and image processing for forward looking infrared (FLIR) imaging systems until 1983. During this time, he held Hughes' Masters, Engineer, and Doctoral Fellowships. In 1983, he joined the faculty of Yale University, New Haven, CT, where he currently is an Associate Professor of Diagnostic Radiology and Electrical Engineering. His research and teaching efforts have been in the areas of computer vision, image processing, and medical imaging. His current specific research interests include the segmentation of deformable objects from both 2-D and 3-D image data, the tracking of nonrigid object motion from both 2-D and 3-D image data, and the integration of processing modules in vision systems, all with a special interest in using these approaches for medical image analysis.

Dr. Duncan is a member of Eta Kappa Nu and Sigma Xi. He is currently an Associate Editor for the IEEE TRANSACTIONS ON MEDICAL IMAGING.