Active Contours for Tracking Distributions

Daniel Freedman, *Member, IEEE,* and Tao Zhang, *Student Member, IEEE*

***Abstract*—A new approach to tracking using active contours is presented. The class of objects to be tracked is assumed to be characterized by a probability distribution over some variable, such as intensity, color, or texture. The goal of the algorithm is to find the region within the current image, such that the sample distribution of the interior of the region most closely matches the model distribution. Two separate criteria for matching distributions are examined, and the curve evolution equations are derived in each case. The flows are shown to perform well in experiments.**

***Index Terms*—Active contours, Bhattacharyya measure, Kullback-Leibler distance, level set method, partial differential equations, visual tracking.**

I. Introduction

THIS PAPER deals with the problem of tracking an object as it moves through a video-stream, based on photometric rather than geometric considerations. Throughout, the term "photometric variable" will be used loosely to mean a quantity such as intensity, color, or texture; photometric variables are distinguished from geometric variables, such as edges. With this in mind, the algorithm may be explained in a straightforward fashion. The class of objects to be tracked is assumed to be characterized by a probability distribution over some photometric variable. In each frame of the video, the algorithm tries to find a region of the image whose interior generates a sample distribution over the relevant variable which most closely matches the model distribution. The goal is to cast this problem into the framework of active contours, and to derive curve flows which optimize the relevant matching criteria. The idea behind the algorithm is illustrated in Fig. 1.

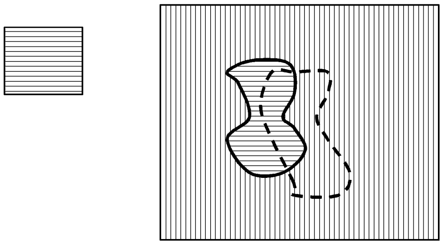


Fig. 1. An illustration of the algorithm. On the left is a schematic representation of the model distribution, here taken to be a texture of horizontal lines. On the right is an image. The dashed line indicates the initial position of the region; within this region, the empirical distribution only partly matches the model distribution, as some of the background (a texture of vertical lines) is contained within the region. Thus, the curve which is the region's boundary will flow to the solid line; the resulting region maximizes the match between empirical and model distributions.

Posing the tracking problem in this way has the advantage of dealing directly with two difficulties that often confound such algorithms. First, the tracker does not rely on edges. Many trackers use edge information exclusively; examples include [1]–[3]. The problems associated with using edges are well-known. For instance, edge-detectors may be inaccurate, leading to the detection of spurious edges; edges may not be detected when contrast between adjacent surfaces fades due to illumination changes; and so on. However, even in the case of ideal edge detection, such algorithms would err in their approach, simply by failing to take into account the rich amounts of information which are available in the photometric variables of the images. For example, much headway may be made in the design of a lip-tracker by noticing that human lips tend to come in a small number of colors. Indeed, color-based methods are often used in special-purpose trackers (e.g., [4], [5] present color-based face-trackers), mostly to excellent effect. The second problem which is directly addressed by this tracker is the difficulty of tracking successfully through cluttered scenes. By posing the problem as one of matching distributions, the tracker has robustness built in from the start, which should allow for a reasonable chance of navigation through clutter.

A reasonable objection may be raised: why not incorporate both photometric and geometric considerations? This, of course, is the eventual goal of this research programme, the first step of which is presented in this paper. However, before attempting to include geometric concerns, it is instructive to see how well a pure photometric tracker can do. A recent general-purpose photometric tracker [6], [7], to be discussed at greater length shortly, demonstrates the ability of such trackers to succeed.

The remainder of the paper is organized as follows. Section II gives a brief review of the literature, focusing on active contour methods and photometric tracking. Section III contains the bulk of the theoretical contribution: after appropriate notation is introduced, two separate flows are derived, corresponding to two density matching measures. These flows are partial integro-differential equations. Section IV is concerned with issues of their implementation on computer. Section V shows the results of using these flows for tracking several sequences, and compares these results with those obtained using geodesic active contours [8] and geodesic active regions [9]. Finally, Section VI concludes.

II. Related Work

The active contour literature is vast, so no attempt will be made to review it comprehensively. The field originated with

Manuscript received October 22, 2001; revised August 14, 2003. This work was supported in part by the U.S. National Science Foundation, under Award IIS-0 133 144. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Patrick Perez.

The authors are with the Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590 USA (e-mail: freedman@cs.rpi.edu; zhangt3@cs.rpi.edu).

Digital Object Identifier 10.1109/TIP.2003.821445

1057-7149/04$20.00 © 2004 IEEE

the snake formulation of Kass, Witkin, and Terzopoulos [1], and many papers in a similar vein followed [10]–[12]. The recent trend has been toward geometric curve evolution [13]–[15], [8], and this paper will follow in that tradition. Many of these recent papers have focused on the novel level-set approach to implementing geometric curve flows [16], which allows for a stable numerical scheme, as well as for changes in topology to be handled without difficulty.

There are several papers in the active contour literature which bear closer relation to the current paper. An early paper which uses probabilistic region-based information for segmentation is that of Zhu and Yuille [17]. More recently, Chan and Vese [18] solve a restricted form of the Mumford–Shah segmentation problem [19], assuming only two regions whose segments are piecewise constant; they have also examined the full-blown Mumford–Shah segmentation [20]. Yezzi *et al.* [21], [22] and Tsai *et al.* [23] also solve a number of segmentation problems, including the full Mumford–Shah segmentation. Noteworthy as well is the paper of Paragios and Deriche [24], which solves a segmentation problem using both boundary and region information. Unlike the approach taken in this paper, the algorithm in [24] assumes that one has access not only to information about the object to be tracked, but about the background as well. Paragios and Deriche have extended this work to the context of motion estimation in [9]. All of the above mentioned papers are similar in spirit to the current paper, in that they use information contained in the interior of the contours within the geometric curve evolution framework. However, the types of information used are quite different, as are the techniques used in deriving the flows.

Finally, it is worth mentioning the work of Comaniciu *et al.* [6], [7], which builds on earlier research of Bradski [25], and which is in a sense most in keeping with the present work. This algorithm attempts to follow a distribution by maximizing the Bhattacharyya measure between a model distribution and an empirical distribution from the current frame. However, this approach is not based on active contours, and one of its major drawbacks is precisely related to this fact: the shape of the object is assumed to be an ellipse. Of course, many objects are not even approximately elliptical (cf. the flexing finger in Section V). The way in which the ellipse translates from frame to frame is the focus of the papers, and is given by so-called “mean-shift analysis”; however, the way in which the ellipse shrinks or enlarges is incorporated in an *ad hoc* fashion. (This was treated in a more rigorous fashion in another paper, [26].) Nonetheless, this tracker performs very well experimentally, and does so in real time.

III. Theory

A. Notation

Let z be the photometric variable of interest. For example, z could be an intensity, color vector, or texture vector. The variable z lives in the space $\mathcal{Z}$, which is assumed to be a Euclidean space of dimension n for some $n \geq 1$. Thus, for intensities, $n = 1$; for colors $n = 3$; and for textures n is the dimension of the output of the relevant filter bank. It is assumed that the class of objects to be tracked is characterized by a model probability density over the variable z, specified by $q(z)$.

The goal is to try to match a sample probability density within a region of the image to the model density. Let $x \in \mathbb{R}^2$ specify the coordinates in the image plane, and let $Z : \mathbb{R}^2 \rightarrow \mathcal{Z}$ be a mapping from the image plane to the space of the photometric variable. Thus, if $\mathcal{Z}$ is the space of intensities, then $Z(x)$ is just a grayscale image; if $\mathcal{Z}$ is the space of colors, then $Z(x)$ is a color image. Denote a region of the image plane by $\omega \subset \mathbb{R}^2$; let $\mathbf{c} = \partial\omega$ be its boundary. We wish to specify $p(z;\omega)$, the sample probability density within the region ω. Let $\theta(z)$ be the n-dimensional Heaviside function, i.e.,

$$\theta(z) = \begin{cases} 1, & z_1, \ldots, z_n \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Then we may write the cumulative distribution function defined inside the region ω as

$$F(z;\omega) = \frac{\int_\omega \theta(z - Z(x))\, dx}{\int_\omega dx}.$$

Thus, the probability density $p(z;\omega)$ is given by

$$p(z;\omega) = \frac{\partial^n F(z;\omega)}{\partial z_1 \ldots \partial z_n} = \frac{\int_\omega \delta(z - Z(x))\, dx}{\int_\omega dx} \equiv \frac{N(z;\omega)}{A(\omega)}$$

where $\delta(\,\cdot\,)$ is the usual n-dimensional delta-function. Note that $A(\omega)$ is the area of the region ω.

B. Density Matching Criteria

The goal is to find the region ω in the image plane such that the sample density $p(z;\omega)$ most closely matches the model density $q(z)$. There are a variety of criteria which can be used to compare the two densities. One obvious candidate is the Kullback–Leibler distance

$$K(\omega) \equiv K(q(\,\cdot\,), p(\cdot;\omega)) = \int q(z) \log\left(\frac{q(z)}{p(z;\omega)}\right) dz.$$

This measure plays an important role in information theory [27]. Note that the Kullback–Leibler distance is not truly a metric, as it is not symmetric in its argument. Nonetheless, it is referred to as a distance, as the smaller it is, the closer are the two distributions.

A second criterion for comparing densities is the Bhattacharyya measure [28]

$$B(\omega) \equiv B(p(\cdot;\omega), q(\,\cdot\,)) = \int \sqrt{p(z;\omega)q(z)}\, dz.$$

This measure varies between 0 and 1, where 0 indicates complete mismatch, and 1 indicates a complete match. In other words, it is an affinity measure, rather than a distance. This is the matching measure used in [6] and [7].

C. A Proposition Concerning Variational Derivatives

The goal of this section is to state the following proposition, which will be useful in later computations. The proposition is given for simply-connected regions, but can be generalized.

Proposition: Let ω be an elementary region of $\mathbb{R}^2$, let $\mathbf{c} = \partial\omega$ be its boundary, and let $\Gamma(\omega) = \int_\omega \mu(x)\,dx$, where μ is C^1. Additionally, let $\delta\Gamma/\delta\mathbf{c}$ be a 2-vector whose ith component is the variational derivative $\delta\Gamma/\delta\mathbf{c}_i$, assuming a particular parameterization for $\mathbf{c}$. Then there exists a parameterization of $\mathbf{c}$ for which

$$\frac{\delta\Gamma}{\delta\mathbf{c}} \propto \mu(\mathbf{c})\mathbf{n}$$

where $\mathbf{n}$ is the normal to $\mathbf{c}$.

This result relies on Green's Theorem, and was demonstrated in the work of Zhu and Yuille [17], as well as that of Chakraborty *et al.* [29].

In the next two sections, we will use this proposition. In the first case case, we will use a gradient descent approach to find an optimum of Γ in terms of $\mathbf{c}$, i.e., a curve for which $\delta\Gamma/\delta\mathbf{c}$ is 0. In the second case, we will use gradient ascent in a precisely analogous manner. As a result, we can safely ignore the positive constant of proportionality.

D. The Kullback-Leibler Flow

Here we wish to minimize $K(\omega)$, so that gradient *descent* is appropriate

$$\frac{\partial\mathbf{c}}{\partial t} = -\frac{\delta K}{\delta\mathbf{c}}$$

Now

$$\begin{aligned} K(\omega) &= \int_{\mathcal{Z}} q(z)\log\frac{q(z)}{p(z;\omega)}\,dz \\ &= \eta - \int_{\mathcal{Z}} q(z)\log p(z;\omega)\,dz \end{aligned}$$

where η is the negative differential entropy of the model distribution, and can be ignored as it does not depend of ω. Now, since $p(z;\omega) = N(z;\omega)/A(\omega)$, we may write

$$K(\omega) = \log(A(\omega)) - \int_{\mathcal{Z}} q(z)\log(N(z;\omega))\,dz$$

so that

$$\frac{\delta K}{\delta\mathbf{c}} = \frac{1}{A}\frac{\delta A}{\delta\mathbf{c}} - \int_{\mathcal{Z}} q(z)\left[\frac{1}{N(z;\omega)}\frac{\delta N(z)}{\delta\mathbf{c}}\right]dz.$$

Now, using the proposition

$$A(\omega) = \int_\omega dx \Rightarrow \frac{\delta A}{\delta\mathbf{c}} = \mathbf{n}.$$

Similarly

$$N(z;\omega) = \int_\omega \delta(z - Z(x))\,dx \Rightarrow \frac{\delta N(z)}{\delta\mathbf{c}} = \delta(z - Z(\mathbf{c}))\mathbf{n}.$$

Thus

$$\begin{aligned} \frac{\delta K}{\delta\mathbf{c}} &= \frac{\mathbf{n}}{A} - \int_{\mathcal{Z}} q(z)\left[\frac{\delta(z - Z(\mathbf{c}))\mathbf{n}}{\int_\omega \delta(z - Z(x))\,dx}\right]dz \\ &= \frac{\mathbf{n}}{A} - \int_{\mathcal{Z}}\left[\frac{q(z)}{N(z)}\delta(z - Z(\mathbf{c}))\,dz\right]\mathbf{n} \\ &= \frac{\mathbf{n}}{A} - \frac{q(Z(\mathbf{c}))}{N(Z(\mathbf{c}))}\mathbf{n} \\ &= \frac{p(Z(\mathbf{c})) - q(Z(\mathbf{c}))}{N(Z(\mathbf{c}))}\mathbf{n}. \end{aligned}$$

Finally, the Kullback–Leibler flow is given by

$$\frac{\partial\mathbf{c}}{\partial t} = \frac{q(Z(\mathbf{c})) - p(Z(\mathbf{c}))}{N(Z(\mathbf{c}))}\mathbf{n}. \qquad (1)$$

There is one point to note in the forgoing derivation. The proposition of Section III-C cannot be used directly to find $\delta N(z)/\delta\mathbf{c}$. The reason is that the proposition requires that the integrand μ be C^1; this condition is obviously not satisfied in the case of $\delta(z - Z(x))$. To get around this problem, the following procedure may be used. For $\delta(\cdot)$, substitute a C^∞ approximation $\delta_\epsilon(\cdot)$, such that $\lim_{\epsilon\to 0}\delta_\epsilon = \delta$. For example, one could use a unit-normalized Gaussian with variance $\epsilon\mathbf{I}$, where $\mathbf{I}$ is the n-dimensional identity matrix. After the calculation is completed, the limit may be taken. Under these assumptions, the results to be derived will not change.

The intuitive meaning of (1) is clear. If the sample density, evaluated at a particular pixel on the boundary, is smaller than the model density, then the curve expands to take in this pixel. This makes sense: by taking in the pixel, the sample density for that value of z will increase, which leads to a better match between the sample density and the model density. Put another way: a lip-tracker based on the Kullback flow will expand to include a reddish pixel, and will contract away from nonreddish pixels.

Finally, it is worth noting that the flow represented by equation (1) is an integro-differential equation. This is due to the fact that the quantities $p(z)$ and $N(z)$ can only be computed by perfoming integrals over the region ω.

E. The Bhattacharyya Flow

As in the case of the simple flow, we wish to maximize the Bhattacharyya measure, and thus we use gradient *ascent*

$$\frac{\partial\mathbf{c}}{\partial t} = \frac{\delta B}{\delta\mathbf{c}}.$$

The Bhattacharyya measure is given by

$$\begin{aligned} B(\omega) &= \int_{\mathcal{Z}} \sqrt{p(z;\omega)q(z)}\,dz \\ &= \int_{\mathcal{Z}} q^{1/2}(z)\frac{N^{1/2}(z;\omega)}{A^{1/2}(\omega)}\,dz \end{aligned}$$

so that

$$\begin{aligned} \frac{\delta B}{\delta\mathbf{c}} &= \int_{\mathcal{Z}} \frac{q^{1/2}(z)}{A(\omega)}\left[A^{1/2}(\omega)\frac{1}{2}N^{-1/2}(z;\omega)\frac{\delta N}{\delta\mathbf{c}}\right. \\ &\qquad \left. -N^{1/2}(z;\omega)\frac{1}{2}A^{-1/2}(\omega)\frac{\delta A}{\delta\mathbf{c}}\right]dz \\ &= \frac{\mathbf{n}}{2A(\omega)}\left[A^{1/2}(\omega)\int_{\mathcal{Z}} q^{1/2}(z)N^{-1/2}(z;\omega)\delta(z - Z(\mathbf{c}))\,dz\right. \\ &\qquad \left. -A^{-1/2}(\omega)\int_{\mathcal{Z}} q^{1/2}(z)N^{1/2}(z;\omega)\,dz\right] \\ &= \frac{\mathbf{n}}{2A(\omega)}\left[A^{1/2}(\omega)q^{1/2}(Z(\mathbf{c}))N^{-1/2}(Z(\mathbf{c}))\right. \\ &\qquad \left. -\int_{\mathcal{Z}} q^{1/2}(z)p^{1/2}(z;\omega)\,dz\right] \\ &= \frac{\mathbf{n}}{2A(\omega)}\left[\frac{q^{1/2}(Z(\mathbf{c}))}{p^{1/2}(Z(\mathbf{c}))} - B(p,q)\right]. \end{aligned}$$

In the foregoing, many of the arguments used in deriving the Kullback–Leibler flow have been recycled. We finally have that the Bhattacharyya flow is given by

$$\frac{\partial \mathbf{c}}{\partial t} = \frac{1}{2A(\omega)} \left[\frac{q^{1/2}(Z(\mathbf{c}))}{p^{1/2}(Z(\mathbf{c}))} - B(p,q) \right] \mathbf{n}. \tag{2}$$

This equation has a similar intuitive understanding as the Kullback flow, except insofar as it is somewhat more aggressive in expanding (since $B(p,q)$ is less than 1, as long as p and q are not equal). Like the Kullback flow, the Bhattacharyya flow is also an integro-differential equation. In this case, the quantities p, A, and B (which depends on p) are computed via integration over ω.

IV. Implementation

There are a variety of issues which arise in the implementation of the Kullback–Leibler and Bhattacharyya flows. It is natural to do this implementation using the level-set method [16]. This framework has gained favor over the last decade within the computer vision community, due to its many advantages over competing approaches (such as marker particles). These include the ability to handle changes in the topology of the curve (splits and merges), the ability to deal with the formation of cusps and corners, which are extremely common in curve evolution, and numerical stability.

In order to convert a curve evolution to the level-set framework, it is necessary for the evolution to be "purely geometric," i.e., for the flow to be entirely in the normal direction. Fortunately, this is already the case in (1) and (2). The level-set function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is chosen so that its 0 level-set corresponds to the curve in question

$$\mathbf{c} = \{x \in \mathbb{R}^2 : \phi(x) = 0\}.$$

In general, we will assume that the points x inside the curve satisfy $\phi(x) < 0$. Given this definition, a curve evolution equation of the form

$$\frac{\partial \mathbf{c}}{\partial t} = \beta \mathbf{n}$$

can be shown to be equivalent [16] to an equation of the form

$$\frac{\partial \phi}{\partial t} + \beta \|\nabla \phi\| = 0.$$

Thus, the Kullback–Leibler flow of (1) becomes

$$\frac{\partial \phi(x,t)}{\partial t} + \frac{q(Z(x)) - p(Z(x))}{N(Z(x))} \|\nabla \phi(x,t)\| = 0$$

while the Bhattacharyya flow of (2) is rendered

$$\frac{\partial \phi(x,t)}{\partial t} + \frac{1}{2A} \left[\frac{q^{1/2}(Z(x))}{p^{1/2}(Z(x))} - B(p,q) \right] \|\nabla \phi(x,t)\| = 0.$$

Note that the integral quanities in the above equations can be computed using our knowledge of ϕ; for example

$$A = \int_{x \in \mathbb{R}^2 : \phi(x) \leq 0} dx.$$

There are two obvious issues which arise when such flows are implemented in the level-set framework. The first pertains to initialization of the level-set function ϕ. A common choice, which we follow here, is to use the signed distance to the intial curve $\mathbf{c}$. A second issue pertains to the size of the time-step allowed. This is dictated by the CFL condition [16], which states that

$$\Delta t \leq \frac{\Delta x}{\max_{x \in \mathbb{R}^2} \beta(x)}$$

where β is the speed of the flow. We typically choose Δt to be about 0.5 times the right side of the above inequality.

Another implementation issue arises from the fact that images are actually discrete-valued, rather than continuous-valued; there is therefore the question of how to best approximate the densities. As in [6], [7], we use histograms. In particular, we can learn the model density q in a straightforward way by taking an image (or multiple images) with an object (objects) drawn from the class of interest, and then finding the histogram within the object. Empirically, q tends to have small support, and is 0 in most $(\approx 99\%)$ of bins.

V. Experiments

Three experiments were performed; their results are summarized in Table I. In all cases, the experiments were run on a Pentium III machine operating at 933 MHz, and using an uncompiled MATLAB implementation. With these parameters, one frame of tracking using either the Kullback–Leibler or Bhattacharyya flow requires approximately 3 min of computation. In the case of the latter two experiments, the results are compared with those obtained by running the geodesic active contour algorithm [8] and the geodesic active region algorithm [9]. These algorithms have been chosen for purposes of comparison for several reasons. The geodesic active contour method is a more rigourous version of a classic technique for tracking and segmentation, namely the elastic snakes. This algorithm only relies on high contrast (edge-based) information, and therefore demonstrates the power of the photometric methods introduced in this paper over pure contour-based methods. By contrast, the geodesic active region method uses information from both the contour and its interior in order to track. Thus, the comparison in this case shows the benefits of using the Kullback–Leibler or Bhattacharyya flows over an important existing photometric method. (Note: no direct speed comparison is presented between the current approach and either the geodesic active contour method or the geodesic active region method; the reason is that our implementations of the latter two trackers have not been optimized for speed, and therefore any comparison would be unfair.)

With regard to comparison with other algorithms, it is also worth commenting on the performance comparison between the proposed flows and two popular methods: the condensation algorithm [2] and mean-shift technique [6], [26]. The condensation algorithm has been shown to have difficulty in following the motion of the finger in the third experiment without a great deal of explicit modeling (e.g., modeling the finger as a robotic arm); for images, see [3]. The mean-shift tracker also cannot track the finger, per se, as it is not designed to deal with flexible shapes;

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS

Experiment	Length (frames)	Flow Used	# Frames Before Tracker Lost Lock
Synthetic	10	Kullback-Leibler	10
Walker	160	Kullback-Leibler	160
Finger	270	Bhattacharyya	270

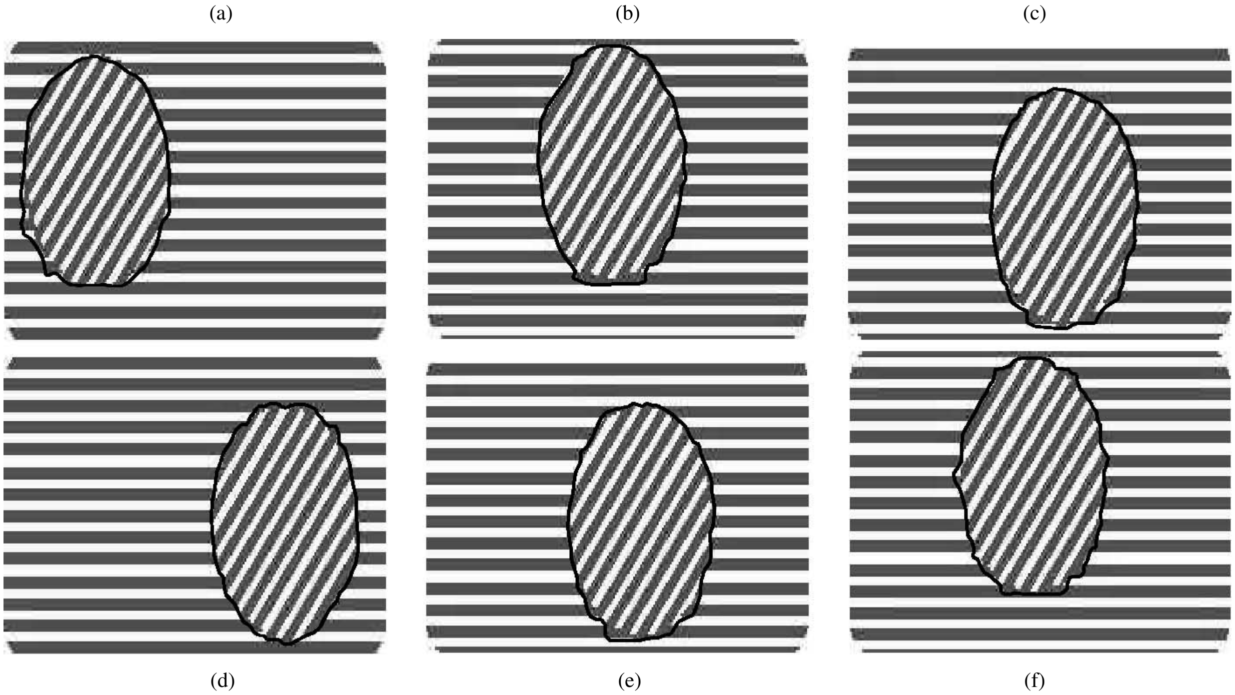


Fig. 2. Tracking a synthetic sequence using the Kullback-Leibler flow. Left to right, and then down: (a) frame 1; (b) frame 3; (c) frame 4; (d) frame 5; (e) frame 6; and (f) frame 7.

rather, the tracking area is always forced to be an ellipse. It could possibly track an elliptical subsection of the finger; however, the goal of tracking is generally to follow an *entire* object as closely as possible.

The first experiment is shown in Fig. 2, and involves a synthetic sequence. The sequence has been designed to demonstrate the ability of the algorithm to track textured regions. The background is composed of horizontal stripes, while the object is composed of diagonal stripes. In this case, it is critical that the photometric variable z be a texture vector, as using intensity or color for the photometric variable would yield the same distribution for the object and the background. A simple texture vector may be chosen based on the directions of (nonzero) intensity gradients in the neighborhood of a pixel; clearly, such a measure will help discriminate between the object and background. The Kullback–Leibler flow is indeed successful in tracking, as is shown in Fig. 2.

The second experiment involves tracking an individual walking; this scene was chosen to illustrate the ability of the algorithm to track through a cluttered scene. In this case, the photometric variable z was taken to be color, specified in HSV coordinates, normalized to run from 0 to 255. The model density q was built as a histogram out of the walker's face taken from a single image (which was not part of the running sequence). The bins were taken to be $8 \times 8 \times 8$, leading to $(256/8)^3 = 32\ 768$ bins. The results of using the Kullback-Leibler flow are shown in Fig. 3; as can be seen, the flow is successful in tracking through the entire 160-frame sequence (=5.3 seconds at 30 Hz). The sequence was also tracked using the method of geodesic active contours [8] and that of geodesic active regions [9]; the results of these experiment are shown in Figs. 5 and 4. Geodesic active contours fail completely in this case; this is due to the poor contrast in the walker's face. As a result, the contour shrinks down to a point by the sixth frame. The geodesic active regions fare better, but completely lose lock by frame 26.

The third experiment involves tracking a finger which both flexes and translates; this scene was chosen to illustrate the ability of the algorithm to track the motion of a nonrigid object. In this case, the photometric variable z was taken to be color, specified in RGB coordinates, normalized to run from 0 to 255. While the HSV color coordinates are generally preferred (and were used in the walker experiment), the tracker does not encounter difficulties through the use of RGB. As in the case

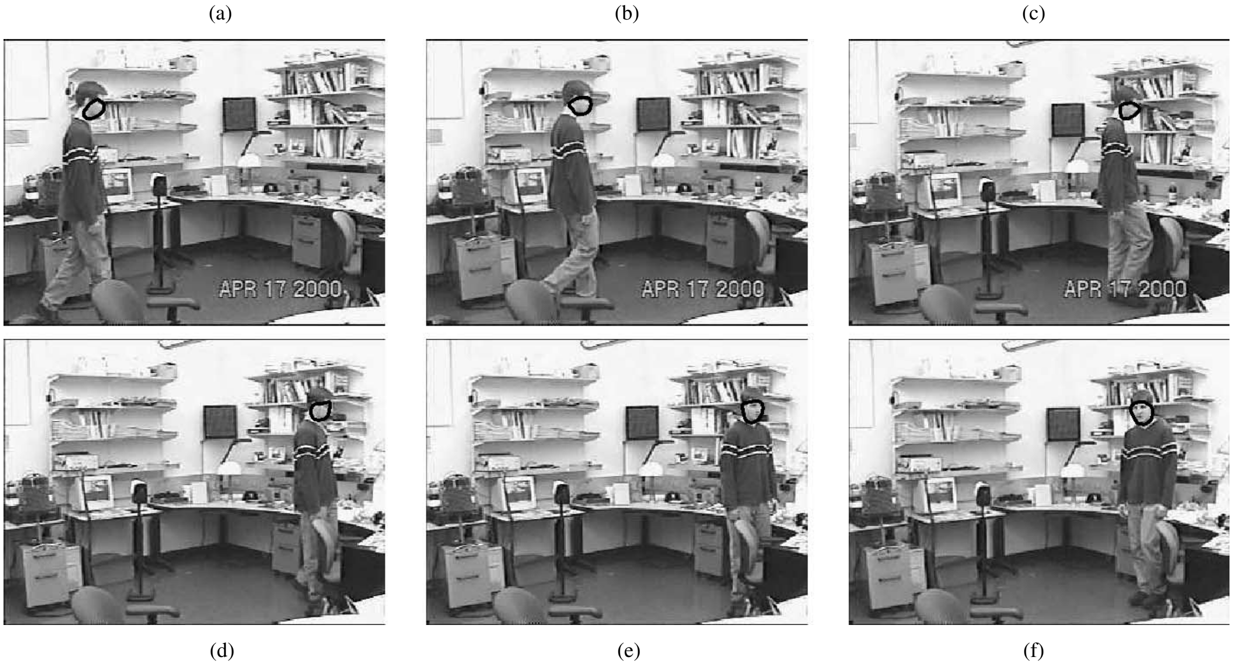


Fig. 3. Tracking a walker using the Kullback-Leibler flow. Left to right, and then down: (a) frame 20; (b) frame 44; (c) frame 91; (d) frame 110; (e) frame 120; and (f) frame 157.

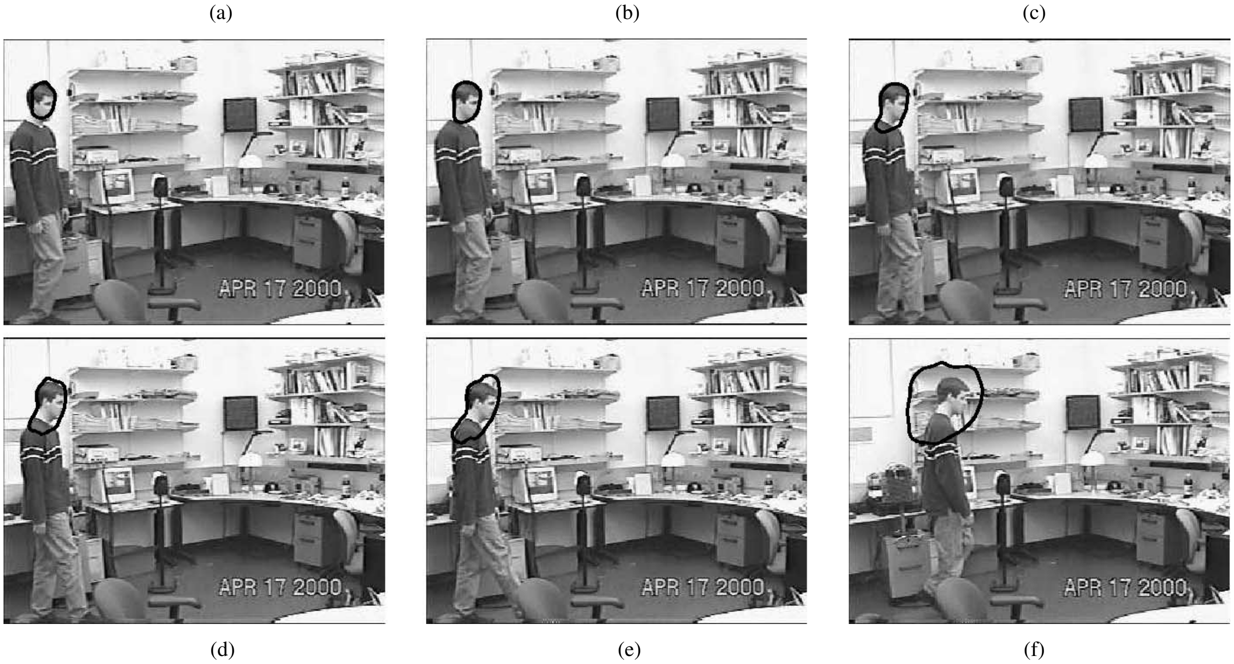
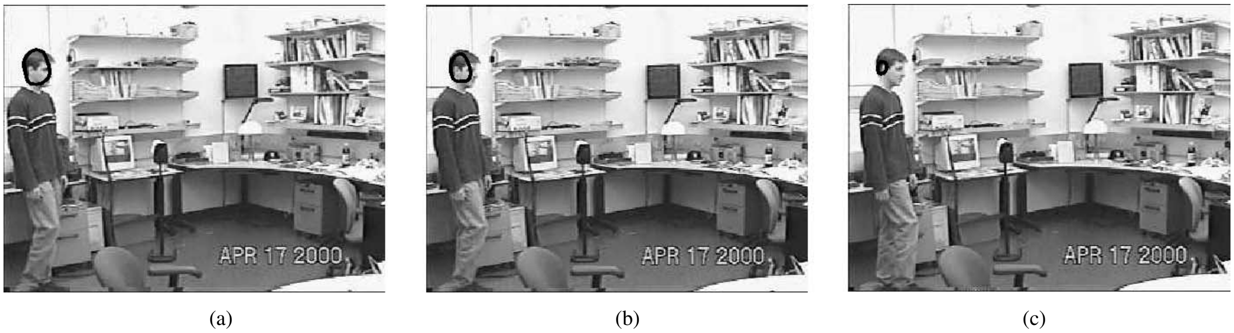


Fig. 4. Tracking a walker using geodesic active regions. Left to right, and then down: (a) frame 3; (b) frame 4; (c) frame 6; (d) frame 7; (e) frame 10; and (f) frame 26.

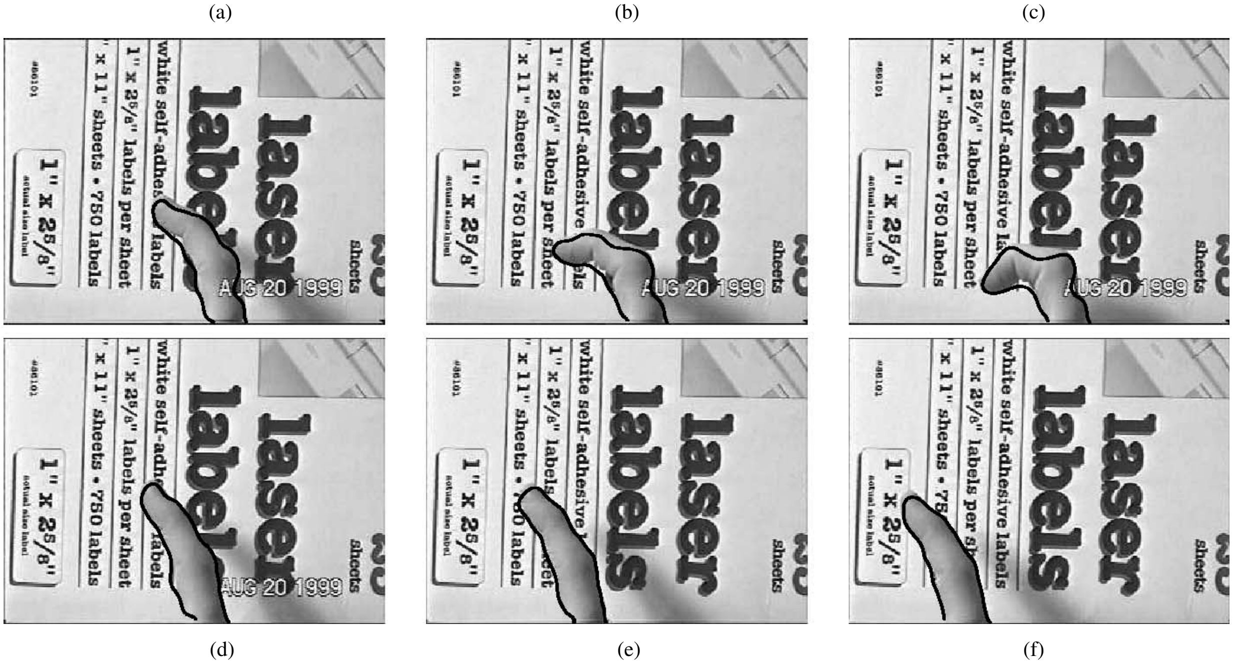
of the walker sequence, the model density q was built as a histogram out of the finger taken from a single image (which was not part of the running sequence). Also, as in the case of the walker sequence, the bins were taken to be $8 \times 8 \times 8$, leading to $(256/8)^3 = 32\,768$ bins. The results of using the Bhattacharyya flow are shown in Fig. 6; as can be seen, the flow is successful in tracking through the entire 270-frame sequence (=9.0 seconds at 30 Hz). The results of the geodesic active contour and geodesic active region experiments are shown in Figs. 8 and 7; once again, these trackers fail where the Bhattacharyya flow succeeds.

There are two interesting artifacts of the flows, which are seen more clearly in the walker sequence than in either of the other two. The first artifact, more easily seen in a video than in still frames, is jitter: despite finding the walker's head correctly in all all frames, there is tendency for the precise shape of the contour to change on a frame by frame basis. This is due to the fact that there are no dynamical considerations incorporated into this



(a) (b) (c)

Fig. 5. Tracking a walker using geodesic active contours. Left to right: (a) frame 2; (b) frame 3; and (c) frame 5.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Tracking a finger using the Bhattacharyya flow. Left to right, and then down: (a) frame 8; (b) frame 14; (c) frame 19; (d) frame 98; (e) frame 118; and (f) frame 143.

algorithm. A second artifact may be labeled "density spill-over," and is illustrated in Fig. 9. Certain colors outside of the object of interest may match those inside the object of interest. This occurs in the case of the walker, as he walks by light colored bookshelves and books, which match the colors of his face. This occurs in five frames of the sequence (frames 96–100), and has no long-term effect, as is illustrated in the final frame of Fig. 9.

VI. Conclusions and Future Work

A new tracking paradigm, based on combining density-matching with active contours, has been presented. Two particular flows were derived, one based on minimizing the Kullback-Leibler distance, the other on maximizing the Bhattacharyya coefficient. The flows have been shown to be effective in practice, in tracking several sequences, and have succeeded where existing methods have failed.

There are several directions for future research. First, more complex measures of z, the photometric variable, will be used. Possibilities include texture vectors (as given by the output of a filter bank), or possibly a neighborhood of texture vectors, in order to effectively capture a Markov random field type of structure. Second, and more importantly, attempts will be made to incorporate geometric considerations into the algorithm. The experiments presented in this paper demonstrate the fact that photometric variables, by themselves, can sometimes be enough to guide a tracker; however, in order to increase both robustness and speed, the use of geometric variables are vital. The challenge is to find a way of incorporating geometry within the framework presented in this paper.

In particular, there are two important shortcomings of the current tracker, which should be addressed. First is the issue of the local nature of the computation: the only bins of the density function which are involved in computing the flow are those bins which contain pixels on the evolving curve. As a result, it is possible to design scenarios in which the flow will certainly fail. For example, imagine the case of an image containing a black disc surrounding by a red annulus of equal areas, and a model density which is half red and half white; if the curve is initially placed inside the red annulus, the curve will converge to the outside of the red annulus, even though the model density ought to contain no black. (On the other hand, it is not clear what

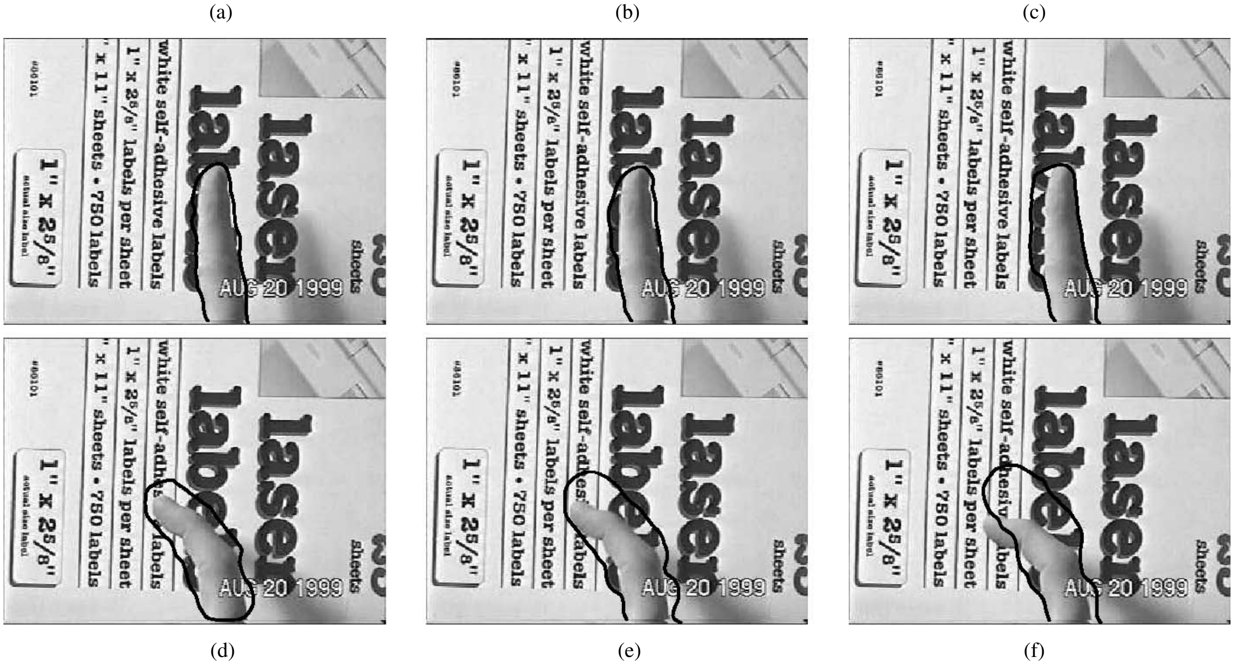


Fig. 7. Tracking a finger using geodesic active regions. Left to right, and then down: (a) frame 1; (b) frame 5; (c) frame 10; (d) frame 30; (e) frame 32; and (f) frame 34.

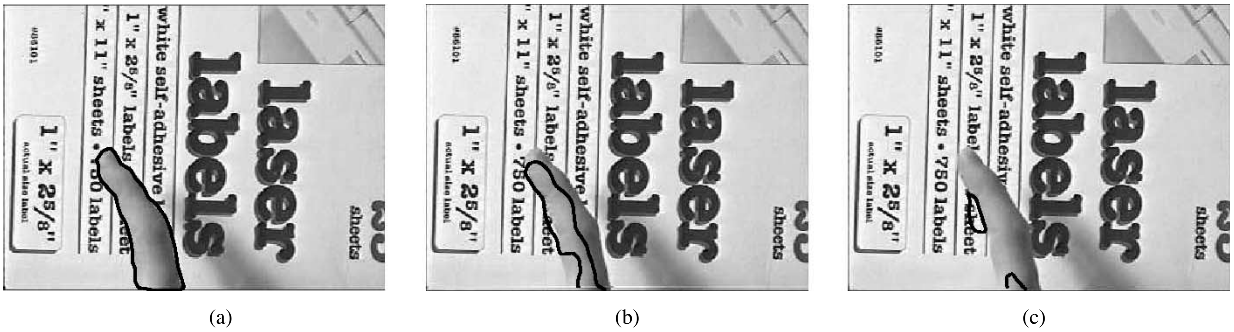


Fig. 8. Tracking a finger using geodesic active contours. In this case, the tracker has been initialized at frame 199. Left to right: (a) frame 199; (b) frame 201; and (c) frame 203.

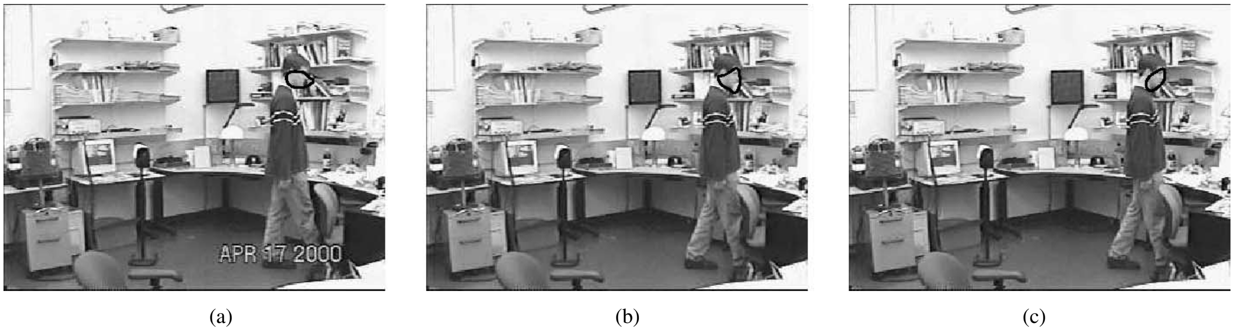


Fig. 9. The density spills over. Left to right: (a) frame 96; (b) frame 100; and (c) frame 101, in which the density has ceased to spill over.

the tracker should converge to in this case, since an object with a sample density matching the model density is not to be found in the image, at least not in the neighborhood of the evolving curve.) Addressing this problem is an important avenue for future research. The second shortcoming relates to the issue of scale. Due to normalization, any two subregions of a homogeneous region will have the same distribution. This problem can be treated by incorporating some prior knowledge of shape into the flows.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” presented at the Int. Conf. Computer Vision, London, U.K., June 1987.

[2] A. Blake and M. Isard, "Condensation-conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[3] D. Freedman and M. Brandstein, "Provably fast algorithms for contour tracking," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 139–144.

[4] J. Yang and A. Waibel, "A real-time face tracer," *Proc. IEEE 3rd WACV*, pp. 142–147, 1996.

[5] N. Oliver, A. P. Pentland, and F. Berard, "Lafter: Lips and face real time tracker," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, 1997, pp. 123–129.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of nonrigid objects using mean shift," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 142–149.

[7] ——, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 564–577, May 2003.

[8] V. Caselles, R. Kimmel, and G. Sapiro, "On geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[9] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 266–280, Mar. 2000.

[10] A. Amini, S. Tehrani, and T. Weymouth, "Using dynamic programming for minimizing the enegy of active contours in the presence of hard constraints," in *Proc. Int. Conf. Computer Vision*, 1988, pp. 95–99.

[11] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 3–20.

[12] G. Xu, E. Segawa, and S. Tsuji, "Robust active contours with insensitive parameters," presented at the Int. Conf. Computer Vision, Berlin, Germany, May 1993.

[13] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numer. Math.*, vol. 66, pp. 1–31, 1993.

[14] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proc. Int. Conf. Computer Vision*, Cambridge, MA, 1995, pp. 810–815.

[15] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, Jan. 1995.

[16] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[17] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.

[18] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, pp. 266–277, Feb. 2001.

[19] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math*, vol. 42, pp. 577–685, 1989.

[20] T. F. Chan and L. A. Vese, "A level-set algorithm for minimizing te mumford-shah functional in image processing," in *Proc. 1st IEEE Workshop Variational and Level Set Methods in Computer Vision*, 2001, pp. 161–168.

[21] A. Yezzi, A. Tsai, and A. S. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *Proc. Int. Conf. Computer Vision*, vol. 2, 1999, pp. 898–903.

[22] ——, "Binary and ternary flows for image segmentation," in *Proc. Int. Conf. Image Processing*, vol. 2, 1999, pp. 1–5.

[23] A. Tsai, A. Yezzi, and A. S. Willsky, "A curve evolution approach to smoothing and segmentation using the mumford-shah functional," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 119–124.

[24] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *Proc. Int. Conf. Computer Vision*, vol. 2, 1999, pp. 926–932.

[25] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. 4th IEEE Workshop Applications of Computer Vision*, 1998, pp. 214–219.

[26] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 281–288, Feb. 2003.

[27] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[28] T. Kailath, "The divergence and the bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Technol.*, vol. COM-15, pp. 52–60, Jan. 1967.

[29] A. Chakraborty, L. H. Staib, and J. S. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Trans. Med. Imag.*, vol. 15, pp. 859–870, Dec. 1996.

Daniel Freedman (M'00) received the B.A. degree in physics from Princeton University, Princeton, NJ, in 1993 and the Ph.D. degree in engineering sciences from Harvard University, Cambridge, MA, in 2000.

He has been Rensselaer Polytechnic Institute, Troy, NY, since 2000, where he is currently an Assistant Professor in the Department of Computer Science. His research interests include computational geometry, computational topology, computer vision, and image processing.

Dr. Freedman is a member of Sigma Xi and the IEEE Computer Society.

Tao Zhang (S'00) received the B.S. degree in automatic control from Harbin Institute of Technology, Harbin, China, in in 1995 and the M.S. degree in automation from Tsinghua University, Tsinghua, China, in 1998. He is currently pursuing the Ph.D. degree in the Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY. His research interests include computer vision, pattern recognition, image analysis, and image processing.

He is a member of the IEEE computer society.