Improving Performance of Distribution Tracking through Background Mismatch

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

Abstract—This paper proposes a new density matching method based on background mismatching for tracking of nonrigid moving objects. The new tracking method extends the idea behind the original density-matching tracker [7], which tracks an object by finding a contour in which the photometric density sampled from the enclosed region most closely matches a model density. This method can be quite sensitive to the initial curve placements and model density. The new method eliminates these sensitivities by adding a second term to the optimization: The *mismatch* between the model density and the density sampled from the background. By maximizing this term, the tracking algorithm becomes significantly more robust in practice. Furthermore, we show the enhanced ability of the algorithm to deal with target objects which possess smooth or diffuse boundaries. The tracker is in the form of a partial differential equation, and is implemented using the level-set framework. Experiments on synthesized images and real video sequences show our proposed methods are effective and robust; the results are compared with several existing methods.

Index Terms—Active contours, density matching, level set method, tracking, PDEs.

1 Introduction

THIS paper deals with the problem of tracking a nonrigid object moving through a cluttered background using photometric features such as intensity, color, or texture. In the original density matching method [7], we assume that the class of objects to be tracked is characterized by a probability distribution over some photometric variable. The tracking task in each frame of the video sequence is to find a region of the image whose sample distribution over the photometric variable most closely matches the model distribution. This method may be labeled *foreground matching*, as the region used for sampling the distribution is the interior of the object of interest (or an estimate of this object).

The algorithm based on foreground matching can be represented in the framework of active contours. A curve evolves via a partial differential equation to the boundary where the matching between the density sampled from the curve-enclosed region and the model density is optimized. Unfortunately, foreground matching can be quite sensitive to both initial curve positions and model densities. Curves evolving under this method often converge to positions corresponding to incorrect local optima. An example of this type of problem is shown in Fig. 1. To attack these sensitivities to initialization and model density, we propose a new algorithm which incorporates *background mismatching*. The idea is simple: Whereas foreground matching attempts to match the region representing the object of interest (i.e., the foreground) to a model, background mismatching seeks to *mismatch* the complement of the region (i.e., the background) to the model. The new methods that result are robust to initial curve positions and density distributions.

The remainder of the paper is organized as follows: Section 2 reviews related literature on tracking. Section 3 introduces the theory of density matching flows and derives the background-mismatching flow. Section 4 analyzes the behavior of different flows in a variety of contexts, and demonstrates the robustness of the new flows to initial curve position. Section 5 discusses the level set method-based algorithm and implementation details. Section 6 shows experimental tracking results on synthesized images and real video sequences and compares our method with existing methods. Section 7 concludes the paper.

- *The authors are with the Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180.*
  E-mail: {zhangt3, freedman}@cs.rpi.edu.

Manuscript received 27 Jan. 2004; revised 14 July 2004; accepted 30 Aug. 2004; published online 13 Dec. 2004.
Recommended for acceptance by S. Sclaroff.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0054-0104.

2 Related Work

Due to wide-ranging applications from video surveillance to medical imaging, there has been substantial research in the area of visual tracking. The seminal work on snakes [9] led researchers into the field of active contours. Many variations and extensions of the original snake method have been proposed, for example [17], [20].

Many tracking methods use edges extensively [8], [18]. These trackers employ a variety of techniques, including Kalman filtering [15] which assumes a linear Gaussian state space model, condensation [1] which uses particle filtering to accommodate a nonlinear non-Gaussian state space model, and deformable templates [23]. Many methods incorporate shape [1] or multiple cues [4] to make the tracking robust. However, when dealing with tracking in clutter, edges alone are often not reliable features due to the presence of spurious edges and noise. Edge-based methods generally require the initial estimate to be very close to the real boundary; if this is not the case, these methods fail simply because they cannot sample from the real boundary.

Currently, there is substantial research on geometric curve evolution. Caselles has proposed the geodesic active contour model [2] to solve image segmentation problems. Paragios and Deriche [14] have extended this idea to motion estimation and proposed the geodesic active region method [13] for tracking by using both boundary and region information. Chan and Vese [3] have developed a curve evolution equation which can achieve good segmentation even when there is no sharp object boundary by solving a simplified form of the Mumford-Shah problem [11]. Yezzi et al. [22], [21] and Tsai et al. [19] have some similar work on segmentation problems. Mansouri has developed a region tracking method [10]. Most of these papers use level-set methods [12] to implement geometric curve flows.

Comaniciu et al. [5], [6] proposes a method based on mean shift analysis to track a distribution by maximizing the Bhattacharyya measure between a model distribution and an empirical distribution. The technique of background weighted histograms [6] has some relation to the idea of background-mismatching. However, the mean shift approach assumes the shape of the object is an ellipse and, therefore, has limited applicability when dealing with deformable objects.

3 Theory

This section establishes the theory behind the new density-matching flows. The reader is sometimes referred to the original density matching paper by the authors [7]. Let $\Omega \subset R^2$ be a bounded region, corresponding to the domain of the image, and let $\omega \subset \Omega$ be the estimated region in which the object of interest resides. Let $z \in \mathbb{R}^n$ be the feature variable; z can be a gray-scale intensity ($n = 1$), color vector ($n = 3$), texture vector (n is the size of the filter-bank), or any other feature which can distinguish the object of interest from the background. Let $Z : \mathbb{R}^2 \rightarrow \mathbb{R}^n$ be the image.

The model density is given by $q(z)$ and is assumed to be learned prior to running the algorithm. One simple method for learning this density is to identify the object of interest within one or more training images, and to compute the histogram of pixels inside this object. Unlike the model density, the sample density (sometimes called the empirical density) must be computed for a given image Z and estimated region $\omega \subset \Omega$; this is achieved by the formula

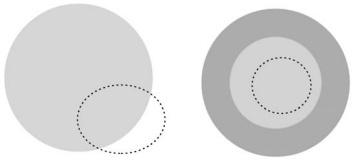

Fig. 1. Two cases where original density matching methods fail. Shaded regions are the target region, dashed curves are initial curves. (a) The target has one color. The curve ends up inside the target region. (b) The target consists of two uniform regions. The curve will shrink and ultimately disappear.

$p(z;\omega) = \int_\omega \delta(z - Z(x))dx / \int_\omega dx$, where $\delta(z)$ is the n-dimensional delta function.

The density matching criteria used in this paper are the Kullback-Leibler distance and the Bhattacharyya measure. The Kullback-Leibler distance is given by $K(\omega) = \int q(z)\log(q(z)/p(z;\omega))dz$. $K(\omega)$ is a nonnegative value measuring the distance or *dissimiliarity* between two probability measures. Smaller $K(\omega)$ values mean smaller distances. (Technically, K is not a true distance/metric.) The Bhattacharyya measure is given by $B(\omega) = \int \sqrt{q(z)p(z;\omega)}dz$. $B(\omega)$ takes on values in $[0,1]$, and measures the *similarity* between densities. Larger $B(\omega)$ values mean smaller distances. Note that B may be easily converted into a metric [6].

The background-mismatching flow is based on the following observation: in most cases, when the estimated region ω matches the target region, the density *mismatching* between the target region and the background $\Omega\backslash\omega$ is maximum. Thus, a curve evolving in the direction which maximizes the mismatch of densities will hopefully match the target. We will therefore try to *minimize* either $-K(\Omega\backslash\omega)$ or $B(\Omega\backslash\omega)$. Such a flow may potentially be even more effective if we combine it with a foreground-matching flow of the type introduced in [7]. These flows try to match the sample density of the estimated region ω to the model density, i.e., to *maximize* either $-K(\omega)$ or $B(\omega)$. A flow which is based on both background-mismatching and foreground-matching is referred to as a *combination flow*.

The combination flows may be derived by finding the variational derivatives of $K(\Omega\backslash\omega)$ and $K(\omega)$ (or $B(\Omega\backslash\omega)$ and $B(\omega)$) with respect to $c = \partial\omega$, and then using gradient descent. Using the methods of [7], the Kullback-Leibler combination flow may be shown to be

$$\frac{\partial c}{\partial t} = \left[\lambda_1 \frac{q(Z(c)) - p(Z(c);\omega)}{N(Z(c);\omega)} + \lambda_2 \frac{q(Z(c)) - p(Z(c);\Omega\backslash\omega)}{N(Z(c);\Omega\backslash\omega)}\right]\mathbf{n}, \quad (1)$$

while the Bhattacharyya combination flow is given by

$$\begin{aligned}\frac{\partial c}{\partial t} = &\left[\frac{\lambda_1}{2A(\omega)}\left(\sqrt{\frac{q(Z(c))}{p(Z(c);\omega)}} - B(\omega)\right)\right.\\ &\left.+\frac{\lambda_2}{2A(\Omega\backslash\omega)}\left(\sqrt{\frac{q(Z(c))}{p(Z(c);\Omega\backslash\omega)}} - B(\Omega\backslash\omega)\right)\right]\mathbf{n},\end{aligned} \quad (2)$$

where $\mathbf{n}$ is the outward-pointing unit normal. In both cases, the first term is the foreground-matching term, and the second term corresponds to background mismatch; the coefficients $\lambda_1, \lambda_2 > 0$ give the relative strength of these terms. There are a variety of possibilities for setting these coefficients. For example, pure background-mismatching is given by $\lambda_1 = 0$. Another possibility is $\lambda_1 = B(\omega)$ and $\lambda_2 = 1 - B(\omega)$, where $B(\omega)$ is the Bhattacharyya measure. This choice is discussed at greater length in Section 4.

4 Discussion of Flow Effectiveness

In this section, we discuss the performance of the background-mismatching flow and foreground-matching flow in four ideal situations, illustrated in Fig. 2. In all cases, we assume that 1) the model density is exactly equal to the sample density of the true object (i.e., the model is perfectly accurate); and 2) the model density and the sample density for the region of the initial estimate have nonoverlapping support. These assumptions are clearly ideal assumptions, but they allow us to make some statements about the efficacy of the flows.

In Fig. 2a, there is no intersection between the initial region and the true object. In this case, curves evolving under both foreground-matching and background-mismatching flows shrink and ultimately disappear. This conclusion is based on the fact that the model density values at each point on the curve are zero, while the sample density values are positive; as a result, according to (1) and (2), the curve may only evolve in the direction of the inward normal at each point. Thus, without modification both the background-mismatching flow and the foreground-matching flow will fail in this situation. In general, however, this situation is unlikely to arise. Since we are in the tracking setting, we expect the initial estimate to overlap at least a small amount with the true object; this expectation is one of the factors that usually distinguishes tracking from segmentation. In the case where this situation does arise, due to a sudden quick movement of the object, the background-mismatching flow may still be successful with some extra implementation details. These are discussed in Section 5.

In Fig. 2b, the initial estimate is contained entirely within the true object. Suppose that the true object is not uniform, but instead consists of the ringed structure shown in Fig. 1b. If the initial estimate is within the inner ring (as in Fig. 1b), then the sample density at each point on the curve will exceed the model density. As a result, under the foreground-matching flow, the curve will shrink inward and ultimately disappear. By contrast, under the background-mismatching flow the curve will expand outward to expel the "foreground material" and will continue doing so until the estimated region coincides with the true object.

In Fig. 2c, the initial estimate and true object overlap with one another, but neither is a subset of the other. Suppose that the true object is homogeneous, in that it contains a single grayscale intensity (color, texture, etc.). In this case, the sample density exceeds the model density at all points on the curve outside of the true object, while the reverse is true for all points within the true object. Thus, under the foreground-matching flow, the curve will do two things: It will contract away from the exterior of the true object, until it is entirely contained within the true object; simultaneously, it will grow within the true object. While both of these motions are correct, the problem occurs when the curve has finished contracting away from the exterior. In this case, the curve will be contained somewhere within the interior of the object, and at each point on

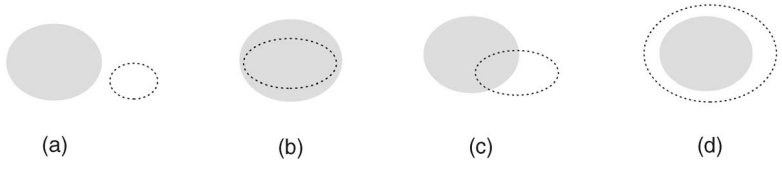

Fig. 2. Relative positions of the true object (shaded region) and the initial position (dotted line).

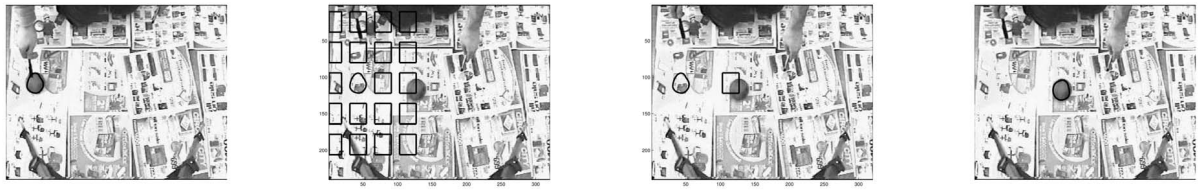

Fig. 3. The curve is initialized as uniformly distributed rectangles around the previous frame's estimate, and the ones with largest Bhattacharyya values are selected as new initial curves (a) frame 305, (b) searching positions, (c) the new initial curve (the rectangle), and (d) tracking result for frame 306.

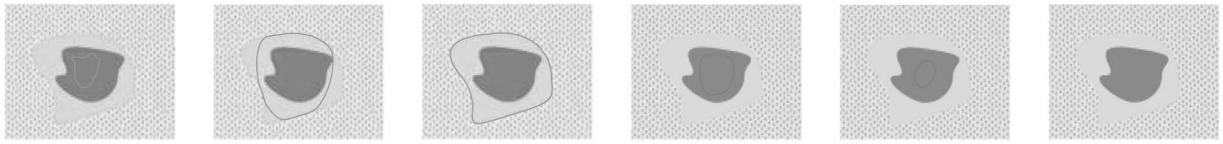

Fig. 4. Tracking in synthesized images. The leftmost three images use background-mismatching; the rightmost three images use foreground-matching. Both flows are based on the KL distance.

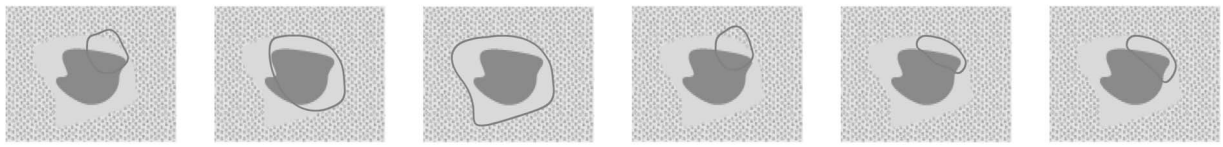

Fig. 5. Tracking in synthesized images. The leftmost three images use background-mismatching; the rightmost three images use foreground-matching. Both flows are based on the KL distance.

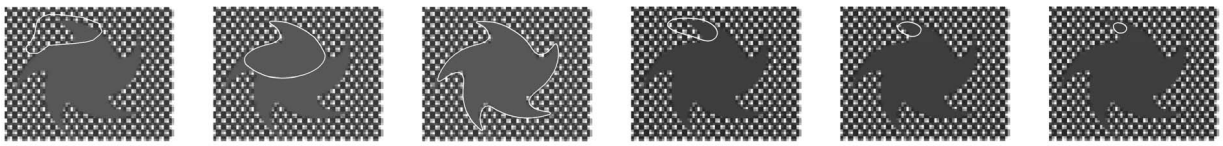

Fig. 6. Tracking in synthesized images. The leftmost three images use background-mismatching; the rightmost three images use foreground-matching. Both flows are based on the Bhattacharyya measure.

the curve, the sample and model densities will exactly match. As a result, the curve will not move, and this is the final position. In fact, there is very little we can say about where, within the true object, the curve will end up. Once again, background-mismatching cures the problem. The curve cannot end up inside the true object, as the mismatch between the model density and background sample density could always be improved by growing the curve until it reaches the boundaries of the true object. Real examples of Figs. 1b and 1c are illustrated in Figs. 4, 5, and 6.

In Fig. 2d, the initial estimate contains the true object. For both foreground-matching or background-mismatching flows, it is obvious that the speeds of all points on the initial curve are in the inward normal direction. As a result, both flows converge to the true object.

Note that both foreground-matching flows and background-mismatching flows are special cases of the combination flow. Given properly selected λ_1 and λ_2, the combination flow works well in each of the cases discussed. Generally, a good heuristic for selecting λ_1 and λ_2 is the following: maintain $\lambda_2 > \lambda_1$ initially to ensure convergence, and let $\lambda_2 < \lambda_1$ when the curve is approaching the target to ensure a good approximation to the true object.

5 Implementation

The level set method [16] is used to implement the curve flows, whether they be background-mismatching or combination flows. The level-set version of (1) is

$$\frac{\partial \phi}{\partial t} = -\left[\lambda_1 \frac{q(Z(x)) - p(Z(x);\omega)}{N(Z(x);\omega)} + \lambda_2 \frac{q(Z(x)) - p(Z(x);\Omega\backslash\omega)}{N(Z(x);\Omega\backslash\omega)}\right]\|\nabla\phi\|.$$

A similar expression can be derived for (2). Densities are approximated by histograms; the computation of the sample density p is made easier by the use of level-sets, as the region ω is easily computed as $\{x : \phi(x) \leq 0\}$. We chose to use a color as the photometric variable; this choice is also made by Comaniciu et al. [6]. Prior to running, the model histogram is found by hand-labeling the true object in a training image, and computing the histogram within this hand-labeled region. In practice, it is somewhat difficult to compute an accurate model density, especially for highly deformable objects moving in a cluttered environment. Thus, the model histogram may be improved by taking several hand-labeled training images, and averaging their histograms.

In general, the final curve estimate for the previous frame is used as the initial estimate within the current frame. However, as has been discussed in Section 4, problems can arise when the true object and the initial estimate do not intersect. In this case, *intersection-checking* may be employed; see Fig. 3. In intersection-checking, the Bhattacharyya measure B between the initial sample density and the model density is used as a measure of intersection. If B is below a threshold, then it is assumed that there is no intersection; instead of using the previous frame's estimate as the initial estimate, the method places $m \times n$ uniformly distributed rectangles centered at the previous frame's estimate and then chooses the k rectangles with

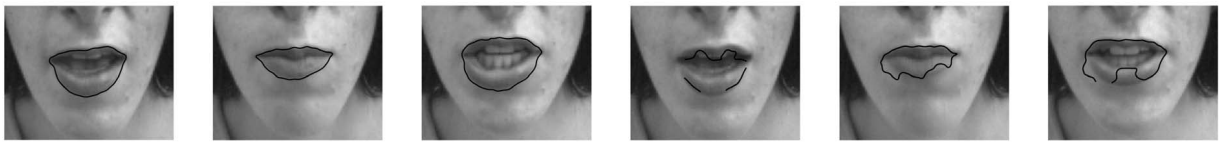

Fig. 7. Lip tracking. The leftmost three images use background-mismatching; the rightmost three images use foreground-matching. Both flows are based on the KL distance.

the highest Bhattacharyya measures. These k rectangles are then used as the initial curve. Example values of some of these parameters are: Bhattacharyya threshold = 0.001, $m = n = 2$, and $k = 1$.

One has to be a little bit careful with intersection-checking. Problems can arise if objects with similar appearance are scattered around the target. In this case, the method might include regions which are not part of the true object.

6 Experimental Results

To test the algorithm, we apply the background-mismatching flow to both synthesized images and real video sequences and compare the results with the following existing trackers.

- Foreground-matching flows [7]. The entire motivation of this paper is to correct the problems associated with the original density-matching flows which did not include background mismatch. It is important to verify that the improvements promised in Section 4 do indeed occur in practice.
- Geodesic active contours [2]. This method, which is essentially a mathematically rigorous version of the elastic snakes [9], is a traditional and often very successful method of tracking. It is therefore a natural source if comparison.
- Geodesic active regions [13]. This method is more sophisticated than geodesic active contours, in that it uses information about both the boundary and the interior of a region. The use of interior information is analogous to that of the density-matching trackers (although the way in which in the information is used is quite different); thus, the comparison is sensible.

Our final comparison shows that the combination flow can outperform the background-mismatching flow in the case when the object boundary is not sharp.

All algorithms have been implemented in Matlab without optimizations under Windows 2000 platform on a Pentium III 450 MHz machine with 192 MB of RAM. The initialization in the first frame of each sequence is done manually. In all cases, the model density has been computed from a single training image. The photometric feature used is color. Trackers based on both the Bhattacharyya measure and Kullback-Leibler distance based have been tested; the results of a single flow are shown unless the results differ.

6.1 Comparison with Foreground-Matching Flow

The advantage of background-mismatching over foreground-matching is illustrated in Figs. 4, 5, 6, and 7. In the first experiment, the object of interest is composed of two uniform regions; we run both background-mismatching and foreground-matching flows for two separate initial estimates, see Figs. 4 and 5. Curves evolving under the background-mismatching flow converge nicely to the true boundary as long as the initial contours intersect the object. By contrast, under the foreground-matching flows curves shrink to disappear when the initial contour is inside the object and are trapped in an incorrect local optimum when the initial contour intersects the object.

In the second experiment, shown in Fig. 6 the object of interest consists of a single uniform region. Curves evolving under the foreground-matching flow stop when the curve is entirely contained within the object, leading to a terrible estimate. Despite the complex shape of the object of interest, the background-mismatching flow performs well.

In the third experiment, we show the efficacy of the background-mismatching flow the in context of lip-tracking; see Fig. 7. The background-mismatching flow tracks the lip with high accuracy, while foreground-matching flow yields results with large errors in most frames, and loses lock at frame 120.

6.2 Comparison with Geodesic Active Contours and Geodesic Active Regions

We now compare the new method with two traditional flow-based trackers. The method of geodesic active contours [2] is very well known, and simple to implement. The more advanced technique of geodesic active regions is presented in a paper by Paragios and Deriche [13]. Like geodesic active contours, this technique uses information about the boundary of the object; unlike the simpler method, it also uses information about the interior. To test its abilities versus those of background-mismatching, we use a cluttered sequence, the "ball" sequence used is shown in Fig. 8. (Note: to implement the geodesic active region method, we select the model parameters $(\alpha, \beta, \gamma, \delta)$ as specified in that paper.)

In Fig. 8, the background-mismatching flow (top row) is compared with both geodesic active contours (middle row) and geodesic active regions (bottom row). Background-mismatching maintains lock for the length of the entire sequence; for nearly all frames, it yields very good results. In a small number of cases, such as that shown in Fig. 8b, the exact boundary of the ball is not detected; this results from the ball and the nearby background having matching color profiles. As a result, color is not a sophisticated enough feature to distinguish the background from the object. When the ball leaves such an area, the tracker recovers the exact position of the ball. Geodesic active contours rely entirely on measurements of intensity contrast; they fail, predictably, due to the high degree of clutter in this scene. More interesting is the failure of the geodesic active regions method; this can be explained as follows. In this method, the initial tracking result is very important in defining a visually consistent model. When the background is cluttered, the initial estimation is not accurate (see Fig. 8i). As a result, the object can not be tracked accurately in this sequence; the size of the error varies with the change in background.

6.3 Background-Mismatching Flow versus Combination Flow

The combination flow achieves the same performance as the background-mismatching flow in the above situations. However, there is one important scenario in which a combination flow is superior to a background-mismatching flow: the situation in which the object has a very smooth boundary. As shown in Fig. 9, even if the initial contour is placed very close to the real boundary, under background-mismatching flows the tracking error is large; while under combination flows, the tracking result is satisfactory. The combination flow works better because the foreground-matching term counteracts the background-mismatching term when the estimated curve approaches the boundary of the true object. However, this increased accuracy comes at some cost, as the combination flow is more computationally intensive.

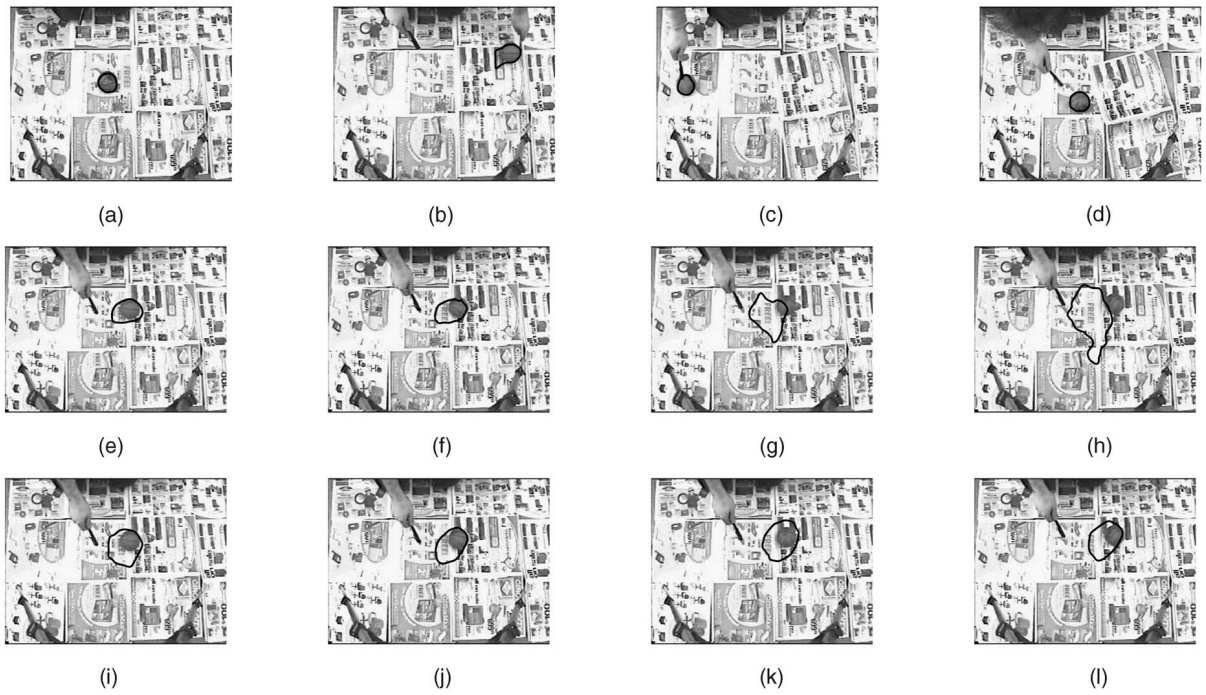

Fig. 8. Tracking a ball. The top row is background-mismatching, based on the Bhattacharyya measure: (a) frame 1, (b) frame 39, (c) frame 305, and (d) frame 509. The middle row is geodesic active contours: (e) frame 28, (f) frame 29, (g) frame 30, and (h) frame 31. The bottom row is geodesic active regions: (i) frame 28, (j) frame 29, (k) frame 30, and (l) frame 31.

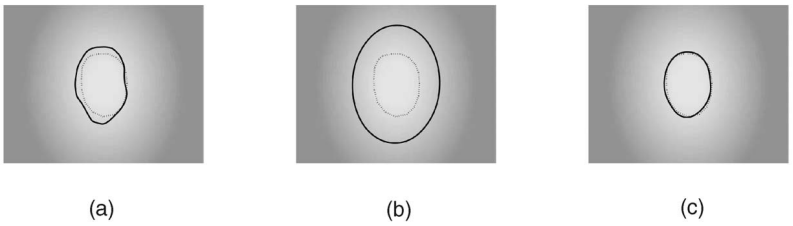

Fig. 9. Tracking an object with smooth boundary, the dashed lines are the real boundaries. (a) The initial curve position, (b) result of the background-mismatching flow, and (c) result of the combination flow. Both flows are based on the Kullback-Leibler distance; the combination flow uses $\lambda_1 = B(\omega), \lambda_2 = 1 - B(\omega)$.

Fig. 10. Woman walking with baby. Frames 1, 80, 226, 329, 421, and 616.

Fig. 11. Swimming fish. Frames 1, 81, 170, 198, 202, and 213.

6.4 More Experiments and Failure of the Proposed Method

In this section, we show the results of two more challenging experiments, as well as some cases which the new method fails. In Fig. 10, we show six frames from a 641 frame surveillance-style sequence (captured at 30 Hz), in which a woman carrying a baby walks through the woods. As can be seen in frame 80, the tracker is occasionally imprecise in capturing the boundaries of the woman (here it is missing the head); however, it always recovers within 1-2 frames. Note also that the tracker is quite good at dealing with changes in scale. In Fig. 11, we show six frames from a 223 frame sequence (captured at 15 Hz) in which a swimming fish is tracked. This sequence is challenging due to the poor contrast between the fish and the background, as well as the clutter (other fish, coral) and the dynamic background. The tracker is successful for the first 197 frames. However, it begins to encounter difficulties in frame 198, as it passes a small object which has a similar intensity

profile. This object, not surprisingly, is incorporated into the estimate of the fish's current location. Problems arise when the object moves further away from the fish, see frames 202 and 213; ultimately, the estimate of the fish's location is quite bad. This sort of failure is quite natural and should be expected, given the framework of density-matching on which the algorithm is based.

7 Conclusions

This paper extends the idea of tracking based on distribution matching by deriving a new flow based on background-mismatching. The new flow is far more robust to initial curve positions as well as variations in the model distribution. In fact, the method does not require the initial curve to be very close to the true object boundary; the contour will converge as long as the initial curve intersects the true object. Experiments show the new method is effective in tracking nonrigid objects in cluttered environments. Furthermore, the flow obtained by combining the background-mismatching flow and the foreground-matching flow improves performance when the object boundary is smooth.

There are two natural directions for future research. The first relates to the choice of the photometric feature. In this paper, we have consistently used color as this feature. However, the photometric variable can be anything which distinguishes the true object from its background. The choice of this feature can be critical to some tracking problems; thus, a more discriminating variable, such as texture, may be appropriate. With high-dimensional variables, it also is important to find reasonable ways of modeling their densities (a histogram is no longer effective). A second direction for research involves incorporating learned shape information into the flows, to make them more robust.

References

[1] A. Blake and M. Isard, "Condensation—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.

[2] V. Caselles, R. Kimmel, and G. Sapiro, "On Geodesic Active Contours," *Int'l J. Computer Vision,* vol. 22, no. 1, pp. 61-79, 1997.

[3] T.F. Chan and L.A. Vese, "Active Contours without Edges," *IEEE Trans. Image Processing,* vol. 10, no. 2, pp. 266-277, 2001.

[4] Y.Q. Chen, T.S. Huang, and Y. Rui, "JPDAF Based HMM for Real-Time Contour Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2001.

[5] D. Comaniciu, "An Algorithm for Data-Driven Bandwidth Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 2, pp. 281-288, Feb. 2003.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-577, May 2003.

[7] D. Freedman and T. Zhang, "Active Contours for Tracking Distributions," *IEEE Trans. Image Processing,* 2004, to appear.

[8] D. Huttenlocher, J. Noh, and W. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes," *Proc. Fourth Int'l Conf. Computer vision,* pp. 93-101, 1993.

[9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Proc. Int'l Conf. Computer Vision,* June 1987.

[10] A.R. Mansouri, "Region Tracking via Level Set PDES without Motion Computation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 947-961, July 2002.

[11] D. Mumford and J. Shah, " Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems," *Comm. Pure Applied Math,* vol. 42, pp. 577-685, 1989.

[12] S. Osher and J.A. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation," *J. Computational Physics,* vol. 79, pp. 12-49, 1988.

[13] N. Paragios and R. Deriche, "Geodesic Active Regions for Motion Estimation and Tracking," *Proc. Int'l Conf. Computer Vision,* vol. 2, pp. 688-694, 1999.

[14] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 3, pp. 266-280, Mar. 2000.

[15] B. Rao, H. Durrant-Whyte, and J. Sheen, "A Fully De-Centralized Multi-Sensor System for Tracking and Surveillance," *Int'l J. Robotics Research,* vol. 12, no. 1, pp. 20-44, 1993.

[16] J.A. Sethian, *Level Set Methods and Fast Marching Methods.* Cambridge Univ. Press, 1999.

[17] D. Terzopoulos and R. Szeliski, "Tracking with Kalman Snakes," *Active Vision,* A. Blake and A. Yuille, eds., pp. 3-20, Cambridge, Mass.: MIT Press, 1992.

[18] K. Toyama and A. Blake, "Probabilistic Tracking in a Metric Space," *Proc. Int'l Conf. Computer Vision,* 2001.

[19] A. Tsai, A. Yezzi, and A.S. Willsky, "A Curve Evolution Approach to Smoothing and Segmentation Using the Mumford-Shah Functional," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 119-124, 2000.

[20] G. Xu, E. Segawa, and S. Tsuji, "Robust Active Contours with Insensitive Parameters," *Proc. Int'l Conf. Computer Vision,* May 1993.

[21] A. Yezzi, A. Tsai, and A.S. Willsky, "Binary and Ternary Flows for Image Segmentation," *Proc. Int'l Conf. Image Processing,* vol. 2, pp. 1-5, 1999.

[22] A. Yezzi, A. Tsai, and A.S. Willsky, "A Statistical Approach to Snakes for Bimodal and Trimodal Imagery," *Proc. Int'l Conf. Computer Vision,* vol. 2, pp. 898-903, 1999.

[23] A. Yuille, D. Cohen, and P. Hallinan, "Feature Extraction from Faces Using Deformable Templates," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 104-109, 1989.