

Tracking Objects Using Density Matching and Shape Priors

Tao Zhang and Daniel Freedman
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180

Abstract

We present a novel method for tracking objects by combining density matching with shape priors. Density matching is a tracking method which operates by maximizing the Bhattacharyya similarity measure between the photometric distribution from an estimated image region and a model photometric distribution. Such trackers can be expressed as PDE-based curve evolutions, which can be implemented using level sets. Shape priors can be combined with this level-set implementation of density matching by representing the shape priors as a series of level sets; a variational approach allows for a natural, parametrization-independent shape term to be derived. Experimental results on real image sequences are shown.

keywords: tracking, shape priors, active contours, density matching, PDEs, level set method

1. Introduction

In recent years, there has been substantial research in active contours and their applications to medical image segmentation and object tracking. Many tracking methods are based either on geometric shape information, such as edges [1, 2], or photometric information, such as color [3, 4]. However, algorithms which do not incorporate both geometric and photometric information are more susceptible to noise, and often perform poorly in cluttered environments.

This paper presents a new approach to the tracking of objects based on density matching and shape priors. Unlike many trackers which try to combine both photometric and geometric priors, the proposed work is posed entirely within the framework of level set and PDEs. (An exception to the previous statement is the work of Paragios and Deriche, such as [5].) The method is appropriate for objects with different color or texture models; by combining both geometric and photometric information, it is relatively immune to clutter.

The paper is organized as follows. The remainder of section 1 is concerned with existing approaches to tracking. Section 2 contains the theory; it derives and explains flows based on both density matching and shape priors.

These flows, which come from variational arguments, are expressed both as curve evolution equations and in terms of level sets. Experimental results on real image sequences and comparisons with the condensation tracker are given in section 3. Section 4 concludes.

1.1. Previous Work

The value of using shape priors has been shown in a variety of tracking contexts. Model-based methods [6] have been used in computer vision for a long time, especially for rigid objects. Deformable templates [7] are an effective and powerful method to model prior shapes and allow for many deformations modes of shapes. However, the modelling of objects by such methods generally requires many parameters and is often done heuristically. Dynamical models, which implicitly contain shape priors, are extensively used in object tracking to improve the robustness by estimating state spaces and predicting possible movements; see, for example, work on Kalman filters [8] and particle filters [1].

Geometric PDEs [9] and variational methods are increasingly used in image segmentation and object tracking. The level set method [10] is an effective framework for implementing these PDEs due to its numerical stability and its ability to cope with topology changes.

The incorporation of shape information into PDE based methods is a new and active research topic; see, for example, [11, 12, 13, 14]. In [11], the family of shape priors is represented by an affine flat in the space of level sets, using principle component analysis. The segmentation method first combines shape priors and image features and uses MAP estimation to derive a global shape, and then evolves active contours under the influence of this global shape. A similar shape representation is used in [15]. In [14], shape is represented by B-splines. The shape energy is defined by the Mahalanobis distance between the given contour and template contour. None of the above methods consider gross shape transformations such as rotation and scaling. Chen's paper [13] defines shape energy as the Euclidean distance between the evolving contour and the shape template. In [12], shape priors are represented by level sets (signed distance transform), and the shape energy is defined

as the integral of the squared distance between the level sets of the evolving curve and the template. Shape priors are learned by a shape-to-area variational approach [16]. Both methods in [13, 16] incorporate Euclidean similarity transformations in the shape energy and estimate parameters by solving differential equations.

2. Theory

Like many existing trackers, our method is based on energy minimization. A tracker cannot track with only shape information; other image-based forces must be included to attract the algorithm to the real contour. Thus, the energy consists of two parts, one term based on image features E_i and a second term based on shape priors E_s :

$$E = \lambda E_i + (1 - \lambda) E_s$$

where $\lambda \in [0, 1]$ defines the relative weight between E_i and E_s . Smaller E means better matching between the estimate and real object. A curve which minimizes the energy E is taken to be solution to the problem.

Section 2.1 describes the energy term E_i based on image features. Section 2.2 proposes an energy term E_s which incorporates a shape prior; partial differential equations for curve flow, based on minimizing these functions through gradient descent, are then derived. Section 2.3 casts this flow in the form of level-set evolution equations, while sections 2.4 and 2.5 consider the impact of euclidean similarity transformations and nonrigid transformations, respectively.

2.1. Image-Based Energy

The most commonly used image feature is the edge-map, see for example [1]; the corresponding image energy is then defined by the distance between the estimated curve and selected feature edges. In this paper, the image features consist of the entire region of intensities (colors, textures) contained within the estimated curve. The distribution over these intensities may be found; the image energy is then based on a measure of closeness between this distribution, and some target distribution which characterizes the object of interest. (The target distribution is usually learned beforehand.) The measure of closeness is generally taken to be an information theoretic measure, such as the Kullback-Leibler distance or the Bhattacharyya measure.

This approach to tracking was used in the mean-shift tracker [3], the trust region-based tracker [17] and prior work of the authors [18, 19]. The approach of the authors differs from the other work in that it allows for arbitrary deformation of the evolving contour, whose motion is described by a partial differential equation. In what follows, this approach is explained briefly, using the Bhattacharyya

measure; for greater detail in both derivations and explanations, as well as the theory based on the Kullback-Leibler distance, the reader is referred to [18, 19].

Let Ω be the entire image plane, and ω the region enclosed by the estimated contour $\partial\omega$ in tracking (see figure 1). Let z be a photometric variable such as intensity, color, or texture; $Z(x)$ is then the image (intensity image, color image, or texture image). For ease of reference, let the true region (i.e. the region we wish our tracker to converge to) be denoted by Φ . With this notation in mind, our goal may be stated as follows: we want to derive a curve evolution equation for the estimated contour $\partial\omega$ under which it will converge to the true contour $\partial\Phi$.

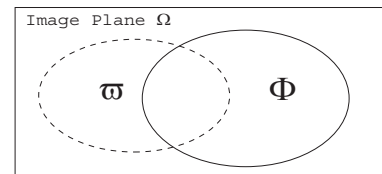


Figure 1: One frame in a image sequence. Ω is the image plane for the frame; Φ is the true region enclosed by the true contour $\partial\Phi$ (solid line); ω is a possible estimated region enclosed by the estimated contour $\partial\omega$ (dashed line).

Our goal is to match the distribution induced by the estimated contour with a model distribution. The model density is given by $q(z)$, and is assumed to be learned prior to running; this density will characterize the interior of the true region Φ . The density of the estimated region, on the other hand, is computed from the image, and is given by

$$p(z; \omega) = \frac{\int_{\omega} \delta(z - Z(x)) dx}{\int_{\omega} dx}$$

where δ is an n -dimensional delta function. This is essentially a continuous version of histogramming, and in practice both q and p are implemented using discrete histograms. In all experiments in this paper, z is taken to be RGB color.

The Bhattacharyya measure can be used to define the distance between distributions $q(z)$ and $p(z; \omega)$:

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

$B(\omega)$ lies in the range $[0, 1]$, and measures the *closeness* of the distributions $q(z)$ and $p(z; \omega)$; that is, larger values of B correspond to more similar distributions. The goal is therefore to *maximize* B . Thus, to convert this to the language of energies, we should set $E_i = -B$. If we let $\mathbf{c}(s)$ be a parametrized version of $\partial\omega$, the boundary of the estimated region, then it can be shown [18] that the

variational derivative of E_i with respect to \mathbf{c} is given be

$$\frac{\delta E_i}{\delta \mathbf{c}} = -\frac{1}{2A(\omega)} \left[B(\omega) - \sqrt{\frac{q(Z(\mathbf{c}(s)))}{p(Z(\mathbf{c}(s)); \omega)}} \right] \mathbf{n}(s)$$

where $A(\omega)$ is the area of the region ω and $\mathbf{n}(s)$ is outward normal at $\mathbf{c}(s)$. E_i can thus be minimized via gradient descent using the following flow:

$$\frac{\partial \mathbf{c}}{\partial t}(s, t) = \frac{1}{2A(\omega)} \left[B(\omega) - \sqrt{\frac{q(Z(\mathbf{c}(s, t)))}{p(Z(\mathbf{c}(s, t)); \omega)}} \right] \mathbf{n}(s, t) \quad (1)$$

While this flow can be quite effective, [19] makes the point that it is also worth considering the region of the image plane, $\omega^c \equiv \Omega - \omega$, which is the complement of the estimated region. We might then try to maximize the *mis-match* between the distribution of this region with the model distribution:

$$E_i(\omega) = B(\omega^c) = \int \sqrt{q(z)p(z; \Omega \setminus \omega)} dz$$

Note that in this case, we wish to minimize B , so we take $E_i = B$, rather than $E_i = -B$. The variational derivative is found in a similar fashion, leading to the flow

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

Of course, flows (1) and (2) may be combined:

$$\begin{aligned} \frac{\partial \mathbf{c}}{\partial t}(s, t) = & \left\{ \frac{\beta}{2A(\omega)} \left[B(\omega) - \sqrt{\frac{q(Z(\mathbf{c}(s, t)))}{p(Z(\mathbf{c}(s, t)); \omega)}} \right] \right. \\ & \left. + \frac{1-\beta}{2A(\omega^c)} \left[\sqrt{\frac{q(Z(\mathbf{c}(s, t)))}{p(Z(\mathbf{c}(s, t)); \omega^c)}} - B(\omega^c) \right] \right\} \mathbf{n}(s, t) \end{aligned} \quad (3)$$

where $\beta \in [0, 1]$ is a parameter to be chosen. In the experiments throughout this paper, we will typically take $\beta = 0$. For convenience, we will generally write equation (3) as

$$\frac{\partial \mathbf{c}(s, t)}{\partial t} = F_i(\mathbf{c}(s, t), t) \mathbf{n}(s, t) \quad (4)$$

where

$$\begin{aligned} F_i(x, t) = & \frac{\beta}{2A(\omega)} \left[B(\omega) - \sqrt{\frac{q(Z(x))}{p(Z(x); \omega)}} \right] \\ & + \frac{1-\beta}{2A(\omega^c)} \left[\sqrt{\frac{q(Z(x))}{p(Z(x); \omega^c)}} - B(\omega^c) \right] \end{aligned}$$

F_i may be thought of as the image-based velocity of the curve flow.

The flow of equation (3) has few restrictions on the distributions and initial positions of estimated contours, has a strong ability to track distributions, and is robust to cluttered environments [19]. However, the accuracy of the target density $q(z)$ affects the accuracy of tracking results. For non-rigid objects moving in highly cluttered environments, it is impossible for $q(z)$ to be an accurate representation of the target distribution in each frame. Shape prior information can therefore be incorporated to make the tracking more accurate.

2.2. Shape Energy

It has long been a practice to incorporate shape priors into tracking algorithms to improve their robustness. Active shape models [20] define the shape priors by using selected landmarks; shape can also be represented by spline curves [6]. In both cases, PCA is used to generate a family of plausible shapes. Correspondence between landmarks or break-points is important in these methods. However, it is preferable not to rely on landmarks, as the choosing of landmarks requires quite a bit of manual labor in training; furthermore, it is not easy to incorporate such landmark-based approaches into a tracker based on PDEs.

Our first attempt at defining a shape energy might run as follows. Suppose that we are given a single curve template, specified parametrically as $\bar{\mathbf{c}}(s)$, and that the curve of interest is given by $\mathbf{c}(s)$. If the parameter s is defined on $[0, 1]$, a natural energy function is

$$E[\mathbf{c}(\cdot)] = \int_0^1 [\mathbf{c}(s) - \bar{\mathbf{c}}(s)]^2 ds$$

Such an energy will indeed achieve its global minimum when $\mathbf{c}(s) = \bar{\mathbf{c}}(s)$ for all values of s . Unfortunately, however, there is a problem with this simple energy: it is dependent upon a parametric specification of both \mathbf{c} and the template $\bar{\mathbf{c}}$. This implies that there must be a reasonable correspondence made between the parameters of \mathbf{c} and $\bar{\mathbf{c}}$; otherwise, the functional E can give meaningless values. For example, suppose that $\mathbf{c}(s) = \bar{\mathbf{c}}(1-s)$; geometrically, then, \mathbf{c} and $\bar{\mathbf{c}}$ represent the same curves. However, they are parametrically distinct, and therefore the value of E will be positive in comparing these curves (and could, potentially, be quite large). This problem is exacerbated by the fact that we often do not reach a global minimum, but instead a local minimum; local minima of this function, reached by gradient descent, may be quite bizarre, assuming that the initial parametric correspondence was not perfect. Furthermore, combining this shape-based criterion with the image-based criterion of the previous section may lead to even more unpredictable results.

Thus, we must find a functional which allows us to match the evolving curve \mathbf{c} with a template curve, and yet does not

rely on a parametric specification of either \mathbf{c} or the template. In order to achieve this goal, let us specify the template as a signed distance function whose zero level set corresponds to the template [10]. That is, let $\bar{\phi} : \mathbb{R}^2 \rightarrow \mathbb{R}$ be such that

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

where $\bar{\mathbf{c}}$ so specified is given as a collection of points. Furthermore, we will assume that $\bar{\phi}(x) < 0$ for all x inside the curve $\bar{\mathbf{c}}$; see figure 2. Clearly, such a specification is non-parametric.

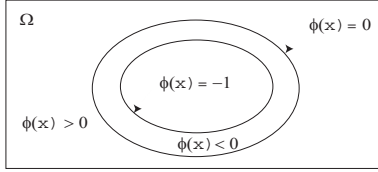


Figure 2: Level set representation. $\phi(x) = 0$ corresponds to the contour; $\phi(x) < 0$ is the inside region; $\phi(x) > 0$ is the outside region. $\phi(x) = -1$ consists of all points whose minimal distance to the contour is -1.

As before, let ω be the region of the plane contained inside the estimated curve \mathbf{c} . In this case, we may formulate the energy functional as

$$E[\mathbf{c}(\cdot)] = \int_{\omega} \bar{\phi}(x) dx$$

Before explaining the rationale behind this function, let us note that it is no way dependent on the particular parametric specification of either the curve \mathbf{c} or the template $\bar{\mathbf{c}}$.

Why will minimizing E drive the curve \mathbf{c} to the template $\bar{\mathbf{c}}$? Recall that $\bar{\phi}(x)$ takes on negative values for x inside the curve $\bar{\mathbf{c}}$. In order to make E as negative as possible, the curve \mathbf{c} should evolve to take on as many of these negative values as possible; the minimum will be achieved when the curve \mathbf{c} surrounds all of the negative values, i.e., coincides exactly with $\bar{\mathbf{c}}$. The great advantage of this functional is that by minimizing it, one drives the curve towards the template, and yet one does not need to worry about any issues related to parametrization.

One can show that

$$\frac{\delta E}{\delta \mathbf{c}} = \bar{\phi}(\mathbf{c}(s)) \mathbf{n}(s)$$

using the same methods of [18]. The corresponding curve evolution equation given by gradient descent is then

$$\frac{\partial \mathbf{c}}{\partial t}(s, t) = -\bar{\phi}(\mathbf{c}(s, t)) \mathbf{n}(s, t) \quad (5)$$

The final curve evolution equation is the combination of the image-based terms of equation (4) and the shape-based

terms of equation (5):

$$\frac{\partial \mathbf{c}}{\partial t}(s, t) = [\lambda F_i(\mathbf{c}(s, t), t) - (1 - \lambda) \bar{\phi}(\mathbf{c}(s, t))] \mathbf{n}(s, t) \quad (6)$$

where $\lambda \in [0, 1]$ is a scalar parameter which gives the relative importance of shape-based information versus photometric information.

2.3. Level Set Formulation

Equation (6) is implemented in the level-set formulation. The straightforward conversion of this equation into level sets yields

$$\frac{\partial \phi}{\partial t}(x, t) = [(1 - \lambda) \bar{\phi}(x) - \lambda F_i(x, t)] \|\nabla \phi(x, t)\|$$

where ϕ is the level-set function corresponding to the evolving curve \mathbf{c} . While there is nothing barring our way from implementing this equation, we may make one small change, in order to achieve a more reasonable extension velocity. Recall that curve evolution equations only inform us of the velocity on the curve itself; whereas, for a proper level-set implementation, we require the velocity throughout the plane. Any extension of the velocity to the remainder of the plane which is smooth is allowed. Thus, we might reformulate the original curve flow of equation (6) as follows:

$$\frac{\partial \mathbf{c}}{\partial t}(s, t) = \{\lambda F_i(\mathbf{c}(s, t), t) - (1 - \lambda) [\bar{\phi}(\mathbf{c}(s, t)) - \phi(\mathbf{c}(s, t))]\} \mathbf{n}(s, t)$$

The only difference between this equation and equation (6) is the presence of the term $\phi(\mathbf{c}(s, t))$. Note that we can add this term with impunity, as by definition of ϕ , $\phi(\mathbf{c}(s, t)) = 0$ for all s, t . While this term does not affect the curve evolution equation, it does change the level set equation to

$$\frac{\partial \phi}{\partial t}(x, t) = [(1 - \lambda) (\bar{\phi}(x) - \phi(x, t)) - \lambda F_i(x, t)] \|\nabla \phi(x, t)\|$$

While both versions of the level-set will result in the same zero level-set, the extension velocity provided in the latter equation is more intuitive. In this equation, the entire level-set function is being driven towards the template's level-set function, rather than just having the zero level-sets match.

2.4. Euclidean Similarity Transformations

Even for a completely rigid object, the flow of equation (5) is problematic. The reason is that the template given by $\bar{\phi}$ is fixed in space, whereas we expect our rigid object to both translate and rotate. (If this were not the case, there would be no point in tracking!) Thus, we must emend the flows of

equation (5) and (6) in order to take this effect into account. We do so by modifying the prior on a frame-by-frame basis, to take into account euclidean similarity transformations.

A euclidean similarity transformation of a point x into a point x' is given by

$$x' = \alpha Ax + T$$

where α is the isotropic scaling factor, A is the rotation matrix, and T is the translation vector. We would like to find an α , A , and T to transform the template $\bar{\phi}$, i.e. to take our new template to be

$$\bar{\phi}_{\alpha,A,T}(x) = \bar{\phi}(\alpha Ax + T)$$

We use the following method to determine these parameters.

We wish to choose α , A , and T to make $\bar{\phi}_{\alpha,A,T}$ as similar to $\phi(x)$ as possible, where $\phi(x)$ is the level set of the previous frame. Let $Q_\phi = \{x : \phi(x) \leq 0\}$. Then we may estimate $\alpha = |Q_\phi|/|Q_{\bar{\phi}}|$, and T by the displacement between the centroids of Q_ϕ and $Q_{\bar{\phi}}$. It remains to determine the rotation matrix A . To do so, use the Procrustes method [21], which is a landmark-based method. Since we do not have landmarks, we check all possible correspondences between landmarks. This is achieved by the following procedure:

- Scale and translate $\bar{\phi}(x) = 0$ by α and T .
- Sample n points uniformly from $\phi(x) = 0$ and $\bar{\phi}(x) = 0$ respectively and denoted by S_ϕ and $S_{\bar{\phi}}$.
- Initialize a correspondence between S_ϕ and $S_{\bar{\phi}}$.
- Use the Procrustes method to estimate the rotation matrix and its error.
- Generate another correspondence for S_ϕ and $S_{\bar{\phi}}$ and repeat step 4 until all correspondences are tried.
- Select the rotation matrix with the smallest error.

Note that there are not that many correspondences to try, since the points in each case lie along a curve; thus, there are at most n correspondences we must test. The accuracy of above estimation is determined by the number of points sampled; in practice, $n = 50$ seems to give good results, and is very fast. Note that with a small modification, α and T can also be estimated by the Procrustes method.

2.5. Nonrigid Transformations

For rigid objects, a single shape model, combined with the euclidean similarity transformations described in section 2.4, is generally sufficient for tracking. For nonrigid objects, a family of shapes must be used to capture the shape changes. The simplest way to achieve this end is to use a

discrete set of shapes, each represented as a level set function: $\{\bar{\phi}_i\}$. The method of section 2.4 may then be applied, at the beginning of each frame. This method allows us to measure an error from fitting each $\bar{\phi}_i$ to the level set from the previous frame ϕ ; the $\bar{\phi}_i$ with the smallest error is chosen as the template for that frame.

Of course, this procedure has a complexity which is linear in the number of prior shapes in the family. Thus, it is critical to ensure that this number is as small as possible; for example, we should not include all training samples. The following procedure allows one to pare down the number of training samples.

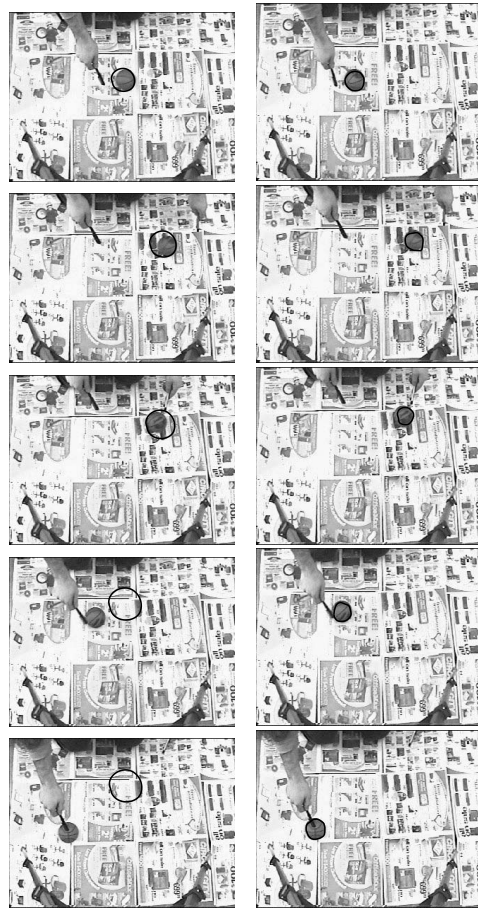


Figure 3: Tracking a ball in clutter. The left column shows results using condensation; the right column shows results using the proposed method.

Let $\{T_1, T_2, \dots, T_n\}$ be n training samples, taken as curves; $\{T_1^i, T_2^i, \dots, T_{i-1}^i, T_{i+1}^i, \dots, T_n^i\}$ are samples which have been aligned to be in correspondence with T_i . The alignment method can be any registration method with good performance, such as the Procrustes method [21] discussed in section 2.4. Let A_i be the interior of T_i and B_j be

the set of interior of T_j^i . The alignment error may be defined in a natural way as $\varepsilon_{ij} = (A_i - B_j) \cup (B_j - A_i)$; ε_{thresh} is some heuristic error threshold. The following procedure may then be used to reduce the number of training samples needed to represent the shape family.

- Select a sample T_i from training samples.
- Align all other samples T_j to T_i and compute the error ε_{ij} . If $\varepsilon_{ij} < \varepsilon_{thresh}$, then remove T_j from the set of training samples.
- Add T_i to the set of shape priors and remove it from the set of training samples.
- Repeat until no training samples remain.

Using the above method, nonrigid objects can be represented by a family of shape priors. However, for some nonrigid objects, this family may be very large.

3. Experimental results

We have tested the tracking algorithm on real image sequences and compared it with the condensation tracker as described in [6]. Both algorithms are implemented in Matlab and run on a 2.0G P4 machine.

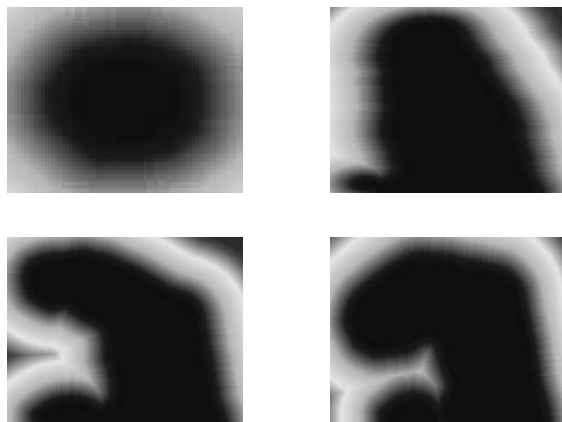


Figure 4: Shape priors. The first one is the signed distance representation of a ball; the remaining three are signed distance representations for a flexing finger.

In the first test sequence, a ball is tracked through a highly cluttered background (see figure 3). The shape prior space in this sequence is a simple circle. The signed distance representation of shape prior is shown in figure 4. The dynamical model for the condensation tracker is simply a uniform distribution over this shape space. Our tracker successfully tracks all the frames, while condensation fails at

frame 69. There is sufficient clutter that several spurious, ball-like arrangements of edge points exist which the condensation tracker takes as the estimated shape.

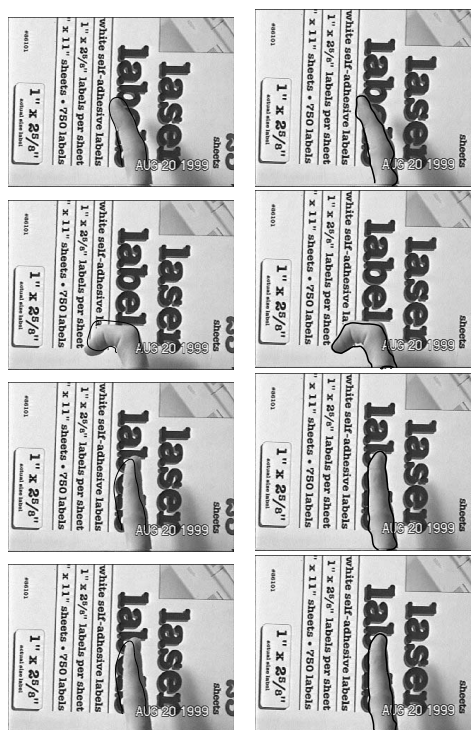


Figure 5: Tracking a translating and flexing finger. The left column shows results using condensation; the right column shows results using the proposed method.

The second sequence involves a finger being tracked as it flexes and translates in a cluttered background (see figure 5). In this case, the shape space is learned from a training sequence; we learn the shape prior using the method given in section 2.5. Some of the members of the shape prior family are shown in figure 4. A PCA-type method is used for condensation. Furthermore, a second order dynamical model is trained on that training sequence. In this sequence, the proposed method tracks well, while condensation yields false estimates (figure 5). When the finger is stationary (see the last two rows in figure 5), condensation still chooses some flex in the finger for its estimate, which is not the true finger position. The proposed method is driven by density; thus, when the finger is stationary, the tracker gives the same contour as that of the previous frame.

There are several deficiencies of the proposed tracker. First, in both experiments condensation is much faster than proposed tracker. Second, because the proposed method does not use edge information, the tracking results do not always rest exactly on the boundary. There is a tendency for the contours obtained to rest somewhere within the real

contour. Finally, the contours are often not very smooth.

4. Conclusions and Future Work

In this paper, we have proposed a tracking method combining density matching and shape priors. The shape priors are represented by level sets. A PDE for curve evolution combining the influences of density matching and of shape priors has been derived. The tracker is seen to track well in cluttered environments; the method outperforms a condensation tracker in the two real image sequences.

One main deficiency of the proposed tracker is its speed: the complexity of the algorithm is linear in the size of the family of shape priors. While one would expect the speed to depend on the size of this family, linearity is probably unacceptable. Future work will focus on reducing this complexity. A second issue which must be addressed is the incorporation of edges, at least in some mild way, in order to get a more accurate localization of the estimated contours.

Acknowledgments

This work was supported in part by the US National Science Foundation, under the award IIS-0133144.

References

- [1] A. Blake and M. Isard, "Condensation - conditional density propagation for visual tracking," *Int. J. of Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.
- [2] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proc. ICCV*, 2001.
- [3] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. CVPR*, vol. 2, pp. 142–149, 2000.
- [4] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Im. Proc.*, vol. 10, no. 2, pp. 266–277, 2001.
- [5] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. PAMI*, vol. 22, no. 3, pp. 266–280, 2000.
- [6] A. Blake and M. Isard, *Active Contours*. Springer, 1998.
- [7] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," in *Proc. CVPR*, pp. 104–109, 1989.
- [8] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," *Active Vision*, pp. 3–20, 1992.
- [9] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
- [10] J. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [11] W. M.E. Leventon and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. CVPR*, vol. 1, pp. 316–323, 2000.
- [12] M. Rousson and N. Paragios, "Shape priors for level set representations," in *Proc. ECCV*, pp. 78–92, 2002.
- [13] Y. Chen and H. D. T. Sheshadri Thiruvenkadam, "On the incorporation of shape priors into geometric active contours," in *IEEE Workshop on Variational and Level Set Methods*, pp. 145–152, 2001.
- [14] D. Cremers, C. Schnorr, and J. Weickert., "Difusion-snakes: combining statistical shape knowledge and image information in a variational framework," in *IEEE Workshop on Variational and Level Set Methods.*, pp. 137–144, 2001.
- [15] A. Tsai and e. Anthony Yezzi, "Model-based curve evolution technique for image segmentation," in *Proc. CVPR*, vol. 1, pp. 463–468, 2001.
- [16] M. R. Nikos Paragios and V. Ramesh, "Matching distance functions: A shape-to-area variational approach for global-to-local registration," in *Proc. ECCV*, pp. 775–789, 2002.
- [17] H.-T. Chen and T. luh Liu, "Trust-region tracker for real time tracking," in *Proc. ICCV*, vol. 2, pp. 717–722, 2001.
- [18] D. Freedman and T. Zhang, "Active contours for tracking distributions." Submitted to *IEEE Trans. Im. Proc.*; under review.
- [19] T. Zhang and D. Freedman, "Improving performance of distribution tracking through background mismatch," tech. rep., Rensselaer Polytechnic Institute, 2003.
- [20] T. Cootes, D. Cooper, C. Taylor, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [21] I.L.Dryden and K.V.Mardia, *Statistical shape analysis*. Chichester New York, 1998.