Probabilistic Tracking with Optimal Scale and Orientation Selection

Hwann-Tzong Chen Tyng-Luh Liu
Institute of Information Science
Academia Sinica
Nankang, Taipei 115, Taiwan

Abstract

We describe a probabilistic framework based on trust-region method to track rigid or non-rigid objects with automatic optimal scale and orientation selection. The approach uses a flexible probability model to represent an object by its salient features such as color or intensity gradient. Depending on the weighting scheme, features will contribute to the distribution differently according to their positions. We adopt a bivariate normal as the weighting function that only features within the induced covariance ellipse are considered. Notice that characterizing an object by a covariance ellipse makes it easier to define its orientation and scale. To perform tracking, a trust-region scheme is carried out for each image frame to detect a distribution similar to the target's accounting for the translation, scale, and orientation factors simultaneously. Unlike other previous work, the optimization process is executed over a continuous space. Consequently, our method is more robust and accurate as demonstrated in the experimental results.

1. Introduction

We aim to develop a general framework for tracking objects in real-time with optimal scale and orientation selection. Our contribution is to establish a trust-region tracker to accomplish the task in the context of probabilistic tracking.

Visual tracking is an important area of research in computer vision. There are trackers that are devised to track contours by the presence of locally detected edges. Isard and Blake introduced CONDENSATION/ICONDENSATION algorithms to track curves in clutter via stochastic analysis and *factor/importance sampling* [7], [8]. Their methods are superior to previous Kalman filter based approaches due to the use of multimodal density. As an alternative to CONDENSATION-based contour tracking techniques, Freedman and Brandstein [5], [6] formulated contour tracking problems as optimization problems. Without assuming a dynamical model, the subset of contour space has to be learned in advance. The tracker then utilizes learned information to find the correct contours among all observations. In [9], Toyama and Blake presented a probabilistic exemplar-based approach for visual tracking. They proposed a *metric mixture* model to combine exemplars in a metric space with a probabilistic treatment.

In [2], Bradski proposed a CAMSHIFT system for use in a perceptual user interface to track face. The method is based on non-parametric technique and *mean shift* to find the peak mode of a color probability distribution. Comaniciu et al. [3] apply mean shift analysis to real-time tracking for non-rigid objects. They measured the similarities between objects using a Bhattacharyya coefficient. Birchfield has proposed an algorithm for head tracking by modeling it as a vertical ellipse with a fixed aspect ratio [1].

Wu and Huang [10] have formulated a non-stationary color tracking problem as a transductive learning problem of training color classifiers. A discriminant-EM algorithm was used to transduce color classifiers and to select a good color space. More recently, they proposed a co-inference approach based on the idea that the process of inference in a higher dimensional state space can be factorized into the process of inference in lower dimensional state spaces in an iterative fashion [11]. A sequential Monte Carlo technique was applied to approximate the co-inference process between the shape and color models.

Our approach relies on probabilistic distributions to characterize rigid or non-rigid objects by their salient features. Since the focus is to track objects with automatic scale and orientation selection, a covariance ellipse model based on bivariate normal distribution is used for the representation. Therefore, to track an object is equivalent to find a similar feature distribution accounting for the factors of translation, rotation and possible non-uniform scaling along the principal axes of the ellipse. Unlike other previous work, e.g., [1], [3], where the scale and orientation are limited only to some pre-determined values, we formulate a *trust-region tracker* to derive an optimal solution. The implication is that by doing the optimization in a continuous and well-scaled space, we are able to derive better tracking performances.

2. Trust-Region Methods with Scaled Norm

2.1. Trust-Region Algorithm

A trust-region method solves an optimization problem iteratively. Its concept can be better understood by considering a typical unconstrained optimization problem,

$$\min_{\mathbf{x}\in\mathbf{V}} f(\mathbf{x}), \tag{1}$$

where $\mathbf{V}$ is a vector space, and f is some objective function to be minimized. Unlike line-search based algorithms, e.g., the *steepest descent*, where at each iterative step the gradient descent direction is the only consideration for finding the next iterate to reduce the value of objective function further. Instead, a trust-region method chooses a more intelligent approach by first constructing a model m to approximate f in a region containing the current iterate, then computing a model minimizer in the region to determine the next iterate.

Essentially, there are three elements of any trust-region methods: (i) *trust-region radius*, to determine the size of a trust region, (ii) *trust-region subproblem*, to approximate a minimizer in the region, and (iii) *trust-region fidelity*, to evaluate the accuracy of an approximating solution. To illustrate, suppose an initial guess $\mathbf{x}_0$ and an initial trust-region radius $\triangle_0 > 0$ are given. Let η_1 and η_2 be some constants satisfying $0 < \eta_1 \leq \eta_2 < 1$. For each iteration $k \geq 0$, we first define an iteration-dependent norm $\|\cdot\|_k$ and iteration-dependent inner product $<\cdot,\cdot>_k$ by

$$\|\mathbf{s}\|_k^2 =< \mathbf{s}, \mathbf{s} >_k \stackrel{\text{def}}{=} < \mathbf{s}, M_k\mathbf{s} >,$$

where $<\cdot,\cdot>$ is the inner product, and M_k is an iteration-dependent matrix. (We will discuss how to determine M_k later.) Then, at iteration k with current iterate $\mathbf{x}_k$, and trust-region radius $\triangle_k$, the following three steps are performed.

1. Trust-region subproblem: We first construct a *model* m_k to approximate f within the current trust region. In this work, a quadratic model is used for the approximation, i.e.,

$$m_k(\mathbf{x}_k + \mathbf{s}) = m_k(\mathbf{x}_k) + < g_k, \mathbf{s} > + \frac{1}{2} < \mathbf{s}, H_k\mathbf{s} >,$$

where $m_k(\mathbf{x}_k) = f(\mathbf{x}_k)$, $g_k = \nabla_{\mathbf{x}} f(\mathbf{x}_k)$, and H_k is the Hessian of f at $\mathbf{x}_k$. For visual tracking with optimal scale and orientation selection, we consider a 5-dimensional vector space $\mathbf{V}$ so that $H_k = \nabla_{\mathbf{xx}} m_k(\mathbf{x}_k)$ is a 5-by-5 symmetric matrix to approximate $\nabla_{\mathbf{xx}} f(\mathbf{x}_k)$. When $H_k \neq 0$, m_k is said to be a second-order model. A trust-region subproblem is then to compute an $\mathbf{s}_k$, where $\|\mathbf{s}_k\|_k \leq \triangle_k$, such that the model m_k is "sufficiently reduced", that is,

$$\min_{\|\mathbf{s}\|_k \leq \triangle_k} \psi_k(\mathbf{s}) =< g_k, \mathbf{s} > + \frac{1}{2} < \mathbf{s}, H_k\mathbf{s} > . \tag{2}$$

2. Trust-region fidelity: After solving the subproblem, the trial point $\mathbf{x}_k + \mathbf{s}_k$ will be tested to see if it is a good candidate for the next iterate. This is evaluated explicitly by the following formula:

$$r_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}.$$

If $r_k \geq \eta_1$, then the trial point is accepted, i.e., $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$. Otherwise, $\mathbf{x}_{k+1} = \mathbf{x}_k$. Since η_1 is a small positive number, the above rule favors a trial point only when the value of objective function f is also reduced. When m_k approximates f well and yields a large r_k, the trust-region radius will be expanded for the next iteration. On the other hand, if r_k is smaller than η_1 or is negative, it suggests that the objective function f is not well approximated by the model function m_k within the current trust region. Therefore, the trust-region radius will be reduced to derive a more appropriate subproblem for the next iteration.

3. Trust-region radius: The new trust-region radius can be updated as follows.

$$\triangle_{k+1} \in \begin{cases} \max\{\alpha_1\|\mathbf{s}_k\|_k, \triangle_k\} & \text{if } r_k \geq \eta_2, \\ \triangle_k & \text{if } r_k \in [\eta_1, \eta_2), \\ \alpha_2\|\mathbf{s}_k\|_k & \text{if } r_k < \eta_1, \end{cases}$$

where following [4] we have used $\eta_1 = 0.05$, $\eta_2 = 0.9$, and $\alpha_1 = 2.5$, $\alpha_2 = 0.25$. The iterative optimization process for (1) will be repeated until the sequence of iterates $\{\mathbf{x}_k\}$ converges.

2.2. Trust-Region Scaled Norm

When an objective function $f(\mathbf{x})$ in (1) has variables whose values are of different orders of magnitude, the optimization problem becomes rather tricky. To resolve such dilemma, trust-region methods provide a convenient way to re-scale the variables. The re-scaling will be done for each iteration k with a *nonsingular* matrix S_k to make sure every trust-region subproblem is solved in a reasonably scaled space. In particular, we have used nonsingular diagonal matrices to scale variables where the diagonal entries correspond to typical values of the respective variables. It follows that the new variables, say $\tilde{\mathbf{x}}$, in the scaled space are derived by $\tilde{\mathbf{x}} = S_k^{-1}\mathbf{x}$. Clearly $\tilde{\mathbf{x}}$ will be of comparable scales after the re-scaling. Nevertheless, as proved in [4], it is not necessary to reformulate a trust-region subproblem using the new variables since re-scaling the variables is equivalent to using an iteration-dependent scaled norm defined by

$$\|\mathbf{s}\|_k^2 =< \mathbf{s}, M_k\mathbf{s} >=< \mathbf{s}, S_k^{-T}S_k^{-1}\mathbf{s} >, \tag{3}$$

where $M_k = S_k^{-T}S_k^{-1}$ is an iteration-dependent matrix.

3. Tracking via Covariance Ellipses

Since our main interest is to track rigid or non-rigid objects using a non-stationary camera, it is convenient to use color distributions to represent objects. Other image features may also be useful. We have been working on combining the color and intensity gradient information for tracking, and the results will be reported somewhere else. To begin with, first divide the RGB color space into n bins, and then define a bin assignment function b by pixel's RGB value as $b : \mathbf{x}_i \mapsto \{1, \ldots, n\}$, where $\mathbf{x}_i$ is any pixel in an image.

3.1. Covariance Ellipse Representation

To model an object reasonably, a weighting scheme is required so that color features at different locations are treated differently. We adopt a bivariate normal distribution to account for translation, scaling and rotation. It is defined by

$$\phi(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} e^{-(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})/2},$$

where $\mathbf{x} = (x_1, x_2)^T$, $\boldsymbol{\mu} = (\mu_1, \mu_2)^T$ is the mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix. Let $\rho = \sigma_{12}/\sigma_1\sigma_2$ and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)^T$. When $|\rho| < 1$, the bivariate normal distribution can be rewritten as

$$\phi(\mathbf{x}; \boldsymbol{\zeta}) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{\varepsilon(\mathbf{x}; \boldsymbol{\zeta})}{2}\right\}, \qquad (4)$$

where we have used $\boldsymbol{\zeta} = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \rho)^T$ to simplify the notation, and

$$\varepsilon(\mathbf{x}; \boldsymbol{\zeta}) = \frac{\left\{\frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho\frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right\}}{1-\rho^2}.$$

Equation (4) implies lines of constant ϕ correspond to constant exponents, i.e., $\varepsilon(\mathbf{x}; \boldsymbol{\zeta}) = constant$. Each such equation represents an ellipse centering at $\boldsymbol{\mu}$. Among them, there is a special one called *covariance ellipse*, $\varepsilon(\mathbf{x}; \boldsymbol{\zeta}) = 1$, where it will be used to compute the color distribution.

Let I^0 be the first image frame and $\boldsymbol{\zeta}^0 = (\boldsymbol{\mu}^0, \boldsymbol{\sigma}^0, \rho^0)^T$. Then, initially, a target centering at $\boldsymbol{\mu}^0$ can be associated with $A(\boldsymbol{\zeta}^0) = \{\mathbf{x} \mid \varepsilon(\mathbf{x}; \boldsymbol{\zeta}^0) \leq 1\}$, the area enclosed by the corresponding covariance ellipse. Furthermore, its color probability distribution, denoted as $p(u; \boldsymbol{\zeta}^0)$, is defined by

$$p(u; \boldsymbol{\zeta}^0) = \frac{1}{C_p} \sum_{\mathbf{x}_i \in A(\boldsymbol{\zeta}^0)} w(\mathbf{x}_i; \boldsymbol{\zeta}^0)\delta(b(\mathbf{x}_i) - u),$$

where δ is the Kronecker delta function and w is a weight function derived from the bivariate normal distribution, i.e.,

$$w(\mathbf{x}_i; \boldsymbol{\zeta}^0) = \exp\left\{-\frac{\varepsilon(\mathbf{x}_i; \boldsymbol{\zeta}^0)}{2}\right\}.$$

To make $p(u; \boldsymbol{\zeta}^0)$ a probability, we have the total weight $C_p = \sum_{\mathbf{x}_i \in A(\boldsymbol{\zeta}^0)} w(\mathbf{x}_i; \boldsymbol{\zeta}^0)$. The notation $p(u; \boldsymbol{\zeta}^0)$ will be abbreviated into $p(u)$ since $\boldsymbol{\zeta}^0$ only describes the target's initial state. Analogously, during tracking, an image area enclosed by $A(\boldsymbol{\zeta})$, its color probability distribution, denoted as $q(u; \boldsymbol{\zeta})$, is

$$q(u; \boldsymbol{\zeta}) = \frac{1}{C_q} \sum_{\mathbf{x}_i \in A(\boldsymbol{\zeta})} w(\mathbf{x}_i; \boldsymbol{\zeta})\delta(b(\mathbf{x}_i) - u),$$

where C_q is the total weight such that $\sum_{u=1}^{n} q(u; \boldsymbol{\zeta}) = 1$.

3.2. A Trust-Region Scheme for Tracking

With the representation, a tracking process for an arbitrary target can be characterized by an evolution dynamics of a covariance ellipse, $\varepsilon(\mathbf{x}; \boldsymbol{\zeta}^t) = 1$. We simply denote the process as $\boldsymbol{\zeta}^0 \rightarrow \boldsymbol{\zeta}^1 \rightarrow \boldsymbol{\zeta}^2 \rightarrow \cdots$. For each image frame I^t, a target is tracked by applying a trust-region method with scaled norm to solve the following optimization problem,

$$\min_{\boldsymbol{\zeta} \in \Omega^t} f(\boldsymbol{\zeta}) = \sum_{u=1}^{n} p(u) \log \frac{p(u)}{q(u; \boldsymbol{\zeta})} + \frac{\lambda}{\sigma_1\sigma_2}, \qquad (5)$$

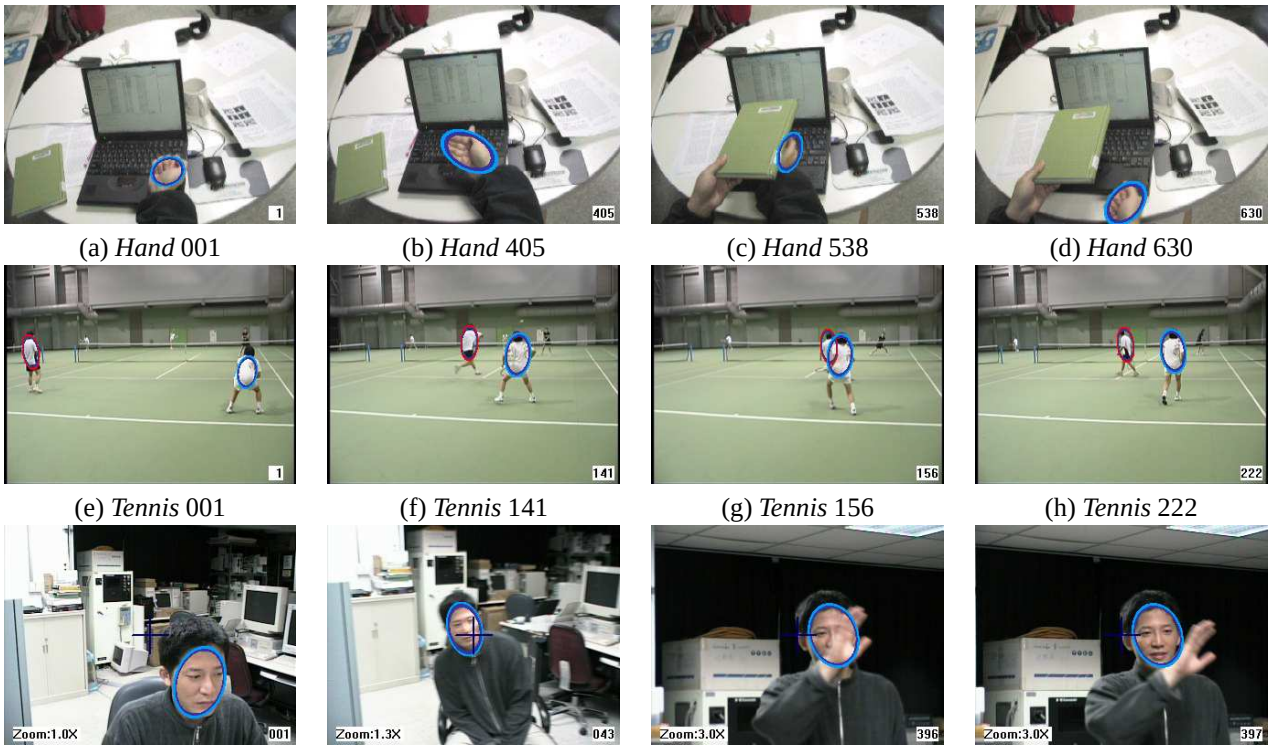
where λ is a parameter, and Ω^t denotes the space consisting of all the possible $\boldsymbol{\zeta}$'s for any combination of translation, scale, and orientation. The objective function in Equation (5) is indeed the familiar *Kullback-Leibler distance* plus a regularization term to favor a larger region when there are several "good" $q(u; \boldsymbol{\zeta})$'s to be considered. The advantage of such modification is most noticeable when tracking an object of monotone color or of uniform pattern.

4. Experimental Results

We have presented a tracking framework using a scale-norm trust-region method. Each target is modeled as a probability distribution within a covariance ellipse. To test our algorithm, a variety of experiments have been carried out, and the outcomes show that a trust-region tracker is very efficient and reliable. Three sets of results are provided (see Figure 1). In each experiment, the RGB space is divided into $16 \times 16 \times 16 = 4096$ bins, and the tracking frame rate is above 30 fps on a Pentium-4 1.5GHz PC. The first two are taken by a hand-held digital video camcorder, and the last one by a pan/tilt/zoom camera. We show that our system can track (i) an object with assorted changes in shape and orientation, (ii) multiple objects with simple interactions, and (iii) an object with automatic pan/tilt/zoom adjustment.

Acknowledgments

This work was supported in part by an NSC grant 90-2213-E-001-016.



(a) *Hand* 001 (b) *Hand* 405 (c) *Hand* 538 (d) *Hand* 630

(e) *Tennis* 001 (f) *Tennis* 141 (g) *Tennis* 156 (h) *Tennis* 222

(i) *Face* 001 (j) *Face* 043 (k) *Face* 396 (l) *Face* 397

Figure 1. (a)-(d) *Hand* **sequence. The experiment demonstrates that the tracker can track a target comfortably, and accounts for changes in target's orientation, shape, and rapid motion along with partial occlusion. (e)-(h)** *Tennis* **sequence. Overlapping occurs during the** 150**th frame to the** 210**th frame, and one player is totally occluded at the** 196**th frame. The tracker continues to track the two players correctly after the overlapping. (i)-(l)** *Face* **sequence. Our system can automatically adjust the pan/tilt/zoom factors so that a target is tracked and displayed appropriately.**

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 232–237, Santa Barbara, CA, 1998.

[2] G. Bradski. Computer vision face tracking for use in a perceptual user interface. In *Intel Technology Journal*, 1998.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, South Carolina, 2000.

[4] A. Conn, N. Gould, and P. Toint. *Trust-Region Methods*. SIAM, Philadelphia, 2000.

[5] D. Freedman and M. Brandstein. Contour tracking in clutter: A subset approach. *Int'l J. Computer Vision*, 38(2):173–186, July 2000.

[6] D. Freedman and M. Brandstein. Provably fast algorithms for contour tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 139–144, Hilton Head Island, South Carolina, 2000.

[7] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *Int'l J. Computer Vision*, 29(1):5–28, August 1998.

[8] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. Fifth European Conf. Computer Vision*, volume 1, pages 893–908, University of Freiburg, Germany, 1998.

[9] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. Eighth IEEE Int'l Conf. Computer Vision*, volume 2, pages 50–57, Vancouver, Canada, 2001.

[10] Y. Wu and T. Huang. Color tracking by transductive learning. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 133–138, Hilton Head Island, South Carolina, 2000.

[11] Y. Wu and T. Huang. A co-inference approach to robust visual tracking. In *Proc. Eighth IEEE Int'l Conf. Computer Vision*, volume 2, pages 26–33, Vancouver, Canada, 2001.