Kernel-Based Object Tracking

Dorin Comaniciu[1] Visvanathan Ramesh[1] Peter Meer[2]

[1]Real-Time Vision and Modeling Department
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540

[2]Electrical and Computer Engineering Department
Rutgers University
94 Brett Road, Piscataway, NJ 08854-8058

Abstract

A new approach toward target representation and localization, the central component in visual tracking of non-rigid objects, is proposed. The feature histogram based target representations are regularized by spatial masking with an isotropic kernel. The masking induces spatially-smooth similarity functions suitable for gradient-based optimization, hence, the target localization problem can be formulated using the basin of attraction of the local maxima. We employ a metric derived from the Bhattacharyya coefficient as similarity measure, and use the mean shift procedure to perform the optimization. In the presented tracking examples the new method successfully coped with camera motion, partial occlusions, clutter, and target scale variations. Integration with motion filters and data association techniques is also discussed. We describe only few of the potential applications: exploitation of background information, Kalman tracking using motion models, and face tracking.

Keywords: non-rigid object tracking; target localization and representation; spatially-smooth similarity function; Bhattacharyya coefficient; face tracking.

1 Introduction

Real-time object tracking is the critical task in many computer vision applications such as surveillance [44, 16, 32], perceptual user interfaces [10], augmented reality [26], smart rooms [39, 75, 47], object-based video compression [11], and driver assistance [34, 4].

Two major components can be distinguished in a typical visual tracker. *Target Representation and Localization* is mostly a bottom-up process which has also to cope with the changes in the appearance of the target. *Filtering and Data Association* is mostly a top-down process dealing with the dynamics of the tracked object, learning of scene priors, and evaluation of different hypotheses. The way the two components are combined and weighted is application dependent and plays a decisive role in the robustness and efficiency of the tracker. For example, face tracking in

a crowded scene relies more on target representation than on target dynamics [21], while in aerial video surveillance, e.g., [74], the target motion and the ego-motion of the camera are the more important components. In real-time applications only a small percentage of the system resources can be allocated for tracking, the rest being required for the preprocessing stages or to high-level tasks such as recognition, trajectory interpretation, and reasoning. Therefore, it is desirable to keep the computational complexity of a tracker as low as possible.

The most abstract formulation of the filtering and data association process is through the *state space approach* for modeling discrete-time dynamic systems [5]. The information characterizing the target is defined by the state sequence $\{\mathbf{x}_k\}_{k=0,1,\ldots}$, whose evolution in time is specified by the dynamic equation $\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k)$. The available measurements $\{\mathbf{z}_k\}_{k=1,\ldots}$ are related to the corresponding states through the measurement equation $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$. In general, both $\mathbf{f}_k$ and $\mathbf{h}_k$ are vector-valued, nonlinear and time-varying functions. Each of the noise sequences, $\{\mathbf{v}_k\}_{k=1,\ldots}$ and $\{\mathbf{n}_k\}_{k=1,\ldots}$ is assumed to be independent and identically distributed (i.i.d.).

The objective of tracking is to estimate the state $\mathbf{x}_k$ given all the measurements $\mathbf{z}_{1:k}$ up that moment, or equivalently to construct the probability density function (pdf) $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. The theoretically optimal solution is provided by the recursive Bayesian filter which solves the problem in two steps. The *prediction* step uses the dynamic equation and the already computed pdf of the state at time $t = k-1$, $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, to derive the prior pdf of the current state, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$. Then, the *update* step employs the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ of the current measurement to compute the posterior pdf $p(\mathbf{x}_k|\mathbf{z}_{1:k})$.

When the noise sequences are Gaussian and $\mathbf{f}_k$ and $\mathbf{h}_k$ are linear functions, the optimal solution is provided by the Kalman filter [5, p.56], which yields the posterior being also Gaussian. (We will return to this topic in Section 6.2.) When the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are nonlinear, by linearization the Extended Kalman Filter (EKF) [5, p.106] is obtained, the posterior density being still modeled as Gaussian. A recent alternative to the EKF is the Unscented Kalman Filter (UKF) [42] which uses a set of discretely sampled points to parameterize the mean and covariance of the posterior density. When the state space is discrete and consists of a finite number of states, Hidden Markov Models (HMM) filters [60] can be applied for tracking. The most general class of filters is represented by particle filters [45], also called bootstrap filters [31], which are based on Monte Carlo integration methods. The current density of the state is represented by a set of

random samples with associated weights and the new density is computed based on these samples and weights (see [23, 3] for reviews). The UKF can be employed to generate proposal distributions for particle filters, in which case the filter is called Unscented Particle Filter (UPF) [54].

When the tracking is performed in a cluttered environment where multiple targets can be present [52], problems related to the validation and association of the measurements arise [5, p.150]. Gating techniques are used to validate only measurements whose predicted probability of appearance is high. After validation, a strategy is needed to associate the measurements with the current targets. In addition to the Nearest Neighbor Filter, which selects the closest measurement, techniques such as Probabilistic Data Association Filter (PDAF) are available for the single target case. The underlying assumption of the PDAF is that for any given target only one measurement is valid, and the other measurements are modeled as random interference, that is, i.i.d. uniformly distributed random variables. The Joint Data Association Filter (JPDAF) [5, p.222], on the other hand, calculates the measurement-to-target association probabilities jointly across all the targets. A different strategy is represented by the Multiple Hypothesis Filter (MHF) [63, 20], [5, p.106] which evaluates the probability that a given target gave rise to a certain measurement sequence. The MHF formulation can be adapted to track the modes of the state density [13]. The data association problem for multiple target particle filtering is presented in [62, 38].

The filtering and association techniques discussed above were applied in computer vision for various tracking scenarios. Boykov and Huttenlocher [9] employed the Kalman filter to track vehicles in an adaptive framework. Rosales and Sclaroff [65] used the Extended Kalman Filter to estimate a 3D object trajectory from 2D image motion. Particle filtering was first introduced in vision as the Condensation algorithm by Isard and Blake [40]. Probabilistic exclusion for tracking multiple objects was discussed in [51]. Wu and Huang developed an algorithm to integrate multiple target clues [76]. Li and Chellappa [48] proposed simultaneous tracking and verification based on particle filters applied to vehicles and faces. Chen *et al*. [15] used the Hidden Markov Model formulation for tracking combined with JPDAF data association. Rui and Chen proposed to track the face contour based on the unscented particle filter [66]. Cham and Rehg [13] applied a variant of MHF for figure tracking.

The emphasis in this paper is on the other component of tracking: target representation and localization. While the filtering and data association have their roots in control theory, algorithms

for target representation and localization are specific to images and related to registration methods [72, 64, 56]. Both target localization and registration maximizes a likelihood type function. The difference is that in tracking, as opposed to registration, only small changes are assumed in the location and appearance of the target in two consecutive frames. This property can be exploited to develop efficient, gradient based localization schemes using the normalized correlation criterion [6]. Since the correlation is sensitive to illumination, Hager and Belhumeur [33] explicitly modeled the geometry and illumination changes. The method was improved by Sclaroff and Isidoro [67] using robust M-estimators. Learning of appearance models by employing a mixture of stable image structure, motion information and an outlier process, was discussed in [41]. In a different approach, Ferrari *et al*. [26] presented an affine tracker based on planar regions and anchor points. Tracking people, which rises many challenges due to the presence of large 3D, non-rigid motion, was extensively analyzed in [36, 1, 30, 73]. Explicit tracking approaches of people [69] are time-consuming and often the simpler blob model [75] or adaptive mixture models [53] are also employed.

The main contribution of the paper is to introduce a new framework for efficient tracking of non-rigid objects. We show that by spatially masking the target with an isotropic kernel, a spatially-smooth similarity function can be defined and the target localization problem is then reduced to a search in the basin of attraction of this function. The smoothness of the similarity function allows application of a gradient optimization method which yields much faster target localization compared with the (optimized) exhaustive search. The similarity between the target model and the target candidates in the next frame is measured using the metric derived from the Bhattacharyya coefficient. In our case the Bhattacharyya coefficient has the meaning of a correlation score. The new target representation and localization method can be integrated with various motion filters and data association techniques. We present tracking experiments in which our method successfully coped with complex camera motion, partial occlusion of the target, presence of significant clutter and large variations in target scale and appearance. We also discuss the integration of background information and Kalman filter based tracking.

The paper is organized as follows. Section 2 discusses issues of target representation and the importance of a spatially-smooth similarity function. Section 3 introduces the metric derived from the Bhattacharyya coefficient. The optimization algorithm is described in Section 4. Experimental results are shown in Section 5. Section 6 presents extensions of the basic algorithm and the new

approach is put in the context of computer vision literature in Section 7.

2 Target Representation

To characterize the target, first a feature space is chosen. The reference *target model* is represented by its pdf q in the feature space. For example, the reference model can be chosen to be the color pdf of the target. Without loss of generality the target model can be considered as centered at the spatial location $\mathbf{0}$. In the subsequent frame a *target candidate* is defined at location $\mathbf{y}$, and is characterized by the pdf $p(\mathbf{y})$. Both pdf-s are to be estimated from the data. To satisfy the low computational cost imposed by real-time processing discrete densities, i.e., m-bin histograms should be used. Thus we have

$$\begin{aligned} &\text{target model:} && \hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\ldots m} && \sum_{u=1}^{m} \hat{q}_u = 1 \\ &\text{target candidate:} && \hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m} && \sum_{u=1}^{m} \hat{p}_u = 1 \,. \end{aligned}$$

The histogram is not the best nonparametric density estimate [68], but it suffices for our purposes. Other discrete density estimates can be also employed.

We will denote by

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \tag{1}$$

a similarity function between $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$. The function $\hat{\rho}(\mathbf{y})$ plays the role of a likelihood and its local maxima in the image indicate the presence of objects in the second frame having representations similar to $\hat{\mathbf{q}}$ defined in the first frame. If only spectral information is used to characterize the target, the similarity function can have large variations for adjacent locations on the image lattice and the spatial information is lost. To find the maxima of such functions, gradient-based optimization procedures are difficult to apply and only an expensive exhaustive search can be used. We regularize the similarity function by masking the objects with an isotropic kernel in the spatial domain. When the kernel weights, carrying continuous spatial information, are used in defining the feature space representations, $\hat{\rho}(\mathbf{y})$ becomes a smooth function in $\mathbf{y}$.

2.1 Target Model

A target is represented by an ellipsoidal region in the image. To eliminate the influence of different target dimensions, all targets are first normalized to a unit circle. This is achieved by independently rescaling the row and column dimensions with h_x and h_y.

Let $\{\mathbf{x}_i^\star\}_{i=1\ldots n}$ be the *normalized* pixel locations in the region defined as the target model. The region is centered at $\mathbf{0}$. An isotropic kernel, with a convex and monotonic decreasing kernel profile $k(x)$[1], assigns smaller weights to pixels farther from the center. Using these weights increases the robustness of the density estimation since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or interference from the background.

The function $b : R^2 \rightarrow \{1 \ldots m\}$ associates to the pixel at location $\mathbf{x}_i^\star$ the index $b(\mathbf{x}_i^\star)$ of its bin in the quantized feature space. The probability of the feature $u = 1 \ldots m$ in the target model is then computed as

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2) \delta \left[b(\mathbf{x}_i^\star) - u \right] \ , \tag{2}$$

where δ is the Kronecker delta function. The normalization constant C is derived by imposing the condition $\sum_{u=1}^{m} \hat{q}_u = 1$, from where

$$C = \frac{1}{\sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2)} \ , \tag{3}$$

since the summation of delta functions for $u = 1 \ldots m$ is equal to one.

2.2 Target Candidates

Let $\{\mathbf{x}_i\}_{i=1\ldots n_h}$ be the *normalized* pixel locations of the target candidate, centered at $\mathbf{y}$ in the current frame. The normalization is inherited from the frame containing the target model. Using the same kernel profile $k(x)$, but with bandwidth h, the probability of the feature $u = 1 \ldots m$ in the target candidate is given by

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right) \delta \left[b(\mathbf{x}_i) - u \right] \ , \tag{4}$$

[1]The profile of a kernel K is defined as a function $k : [0, \infty) \rightarrow R$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$.

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|^2)} \tag{5}$$

is the normalization constant. Note that C_h does not depend on $\mathbf{y}$, since the pixel locations $\mathbf{x}_i$ are organized in a regular lattice and $\mathbf{y}$ is one of the lattice nodes. Therefore, C_h can be precalculated for a given kernel and different values of h. The bandwidth h defines the scale of the target candidate, i.e., the number of pixels considered in the localization process.

2.3 Similarity Function Smoothness

The similarity function (1) inherits the properties of the kernel profile $k(x)$ when the target model and candidate are represented according to (2) and (4). A differentiable kernel profile yields a differentiable similarity function and efficient gradient-based optimizations procedures can be used for finding its maxima. The presence of the continuous kernel introduces an interpolation process between the locations on the image lattice. The employed target representations do not restrict the way similarity is measured and various functions can be used for ρ. See [59] for an experimental evaluation of different histogram similarity measures.

3 Metric based on Bhattacharyya Coefficient

The similarity function defines a distance among target model and candidates. To accommodate comparisons among various targets, this distance should have a metric structure. We define the distance between two discrete distributions as

$$d(\mathbf{y}) = \sqrt{1 - \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right]}\,, \tag{6}$$

where we chose

$$\hat{\rho}(\mathbf{y}) \equiv \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u}\,, \tag{7}$$

the sample estimate of the Bhattacharyya coefficient between $\mathbf{p}$ and $\mathbf{q}$ [43].

The Bhattacharyya coefficient is a divergence-type measure [49] which has a straightforward geometric interpretation. It is the cosine of the angle between the m-dimensional unit vectors $\left(\sqrt{\hat{p}_1}, \ldots, \sqrt{\hat{p}_m}\right)^\top$ and $\left(\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m}\right)^\top$. The fact that $\mathbf{p}$ and $\mathbf{q}$ are distributions is thus explicitly

taken into account by representing them on the unit hypersphere. At the same time we can interpret (7) as the (normalized) correlation between the vectors $\left(\sqrt{\hat{p}_1}, \ldots, \sqrt{\hat{p}_m}\right)^\top$ and $\left(\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m}\right)^\top$. Properties of the Bhattacharyya coefficient such as its relation to the Fisher measure of information, quality of the sample estimate, and explicit forms for various distributions are given in [22, 43].

The statistical measure (6) has several desirable properties:

1. It imposes a metric structure (see Appendix). The Bhattacharyya distance [28, p.99] or Kullback divergence [19, p.18] are not metrics since they violate at least one of the distance axioms.

2. It has a clear geometric interpretation. Note that the L_p histogram metrics (including histogram intersection [71]) do not enforce the conditions $\sum_{u=1}^{m} \hat{q}_u = 1$ and $\sum_{u=1}^{m} \hat{p}_u = 1$.

3. It uses discrete densities, and therefore it is invariant to the scale of the target (up to quantization effects).

4. It is valid for arbitrary distributions, thus being superior to the Fisher linear discriminant, which yields useful results only for distributions that are separated by the mean-difference [28, p.132].

5. It approximates the chi-squared statistic, while avoiding the singularity problem of the chi-square test when comparing empty histogram bins [2].

Divergence based measures were already used in computer vision. The Chernoff and Bhattacharyya bounds have been employed in [46] to determine the effectiveness of edge detectors. The Kullback divergence between joint distribution and product of marginals (e.g., the mutual information) has been used in [72] for registration. Information theoretic measures for target distinctness were discussed in [29].

4 Target Localization

To find the location corresponding to the target in the current frame, the distance (6) should be minimized as a function of $\mathbf{y}$. The localization procedure starts from the position of the target in

the previous frame (the model) and searches in the neighborhood. Since our distance function is smooth, the procedure uses gradient information which is provided by the mean shift vector [17]. More involved optimizations based on the Hessian of (6) can be applied [58].

Color information was chosen as the target feature, however, the same framework can be used for texture and edges, or any combination of them. In the sequel it is assumed that the following information is available: (a) detection and localization in the initial frame of the objects to track (target models) [50, 8]; (b) periodic analysis of each object to account for possible updates of the target models due to significant changes in color [53].

4.1 Distance Minimization

Minimizing the distance (6) is equivalent to maximizing the Bhattacharyya coefficient $\hat{\rho}(\mathbf{y})$. The search for the new target location in the current frame starts at the location $\hat{\mathbf{y}}_0$ of the target in the previous frame. Thus, the probabilities $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$ of the target candidate at location $\hat{\mathbf{y}}_0$ in the current frame have to be computed first. Using Taylor expansion around the values $\hat{p}_u(\hat{\mathbf{y}}_0)$, the linear approximation of the Bhattacharyya coefficient (7) is obtained after some manipulations as

$$\rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\mathbf{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\ . \tag{8}$$

The approximation is satisfactory when the target candidate $\{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}$ does not change drastically from the initial $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$, which is most often a valid assumption between consecutive frames. The condition $\hat{p}_u(\hat{\mathbf{y}}_0) > 0$ (or some small threshold) for all $u = 1\ldots m$, can always be enforced by not using the feature values in violation. Recalling (4) results in

$$\rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right) \tag{9}$$

where

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\delta\left[b(\mathbf{x}_i) - u\right]\ . \tag{10}$$

Thus, to minimize the distance (6), the second term in (9) has to be maximized, the first term being independent of $\mathbf{y}$. Observe that the second term represents the density estimate computed with kernel profile $k(x)$ at $\mathbf{y}$ in the current frame, with the data being weighted by w_i (10). The mode of this density in the local neighborhood is the sought maximum which can be found employing

the mean shift procedure [17]. In this procedure the kernel is recursively moved from the current location $\hat{\mathbf{y}}_0$ to the new location $\hat{\mathbf{y}}_1$ according to the relation

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)} \tag{11}$$

where $g(x) = -k'(x)$, assuming that the derivative of $k(x)$ exists for all $x \in [0, \infty)$, except for a finite set of points. The complete target localization algorithm is presented below.

Bhattacharyya Coefficient $\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]$ *Maximization*

Given:

the target model $\{\hat{q}_u\}_{u=1...m}$ and its location $\hat{\mathbf{y}}_0$ in the previous frame.

1. Initialize the location of the target in the current frame with $\hat{\mathbf{y}}_0$, compute $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1...m}$, and evaluate
$$\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u}\,.$$
2. Derive the weights $\{w_i\}_{i=1...n_h}$ according to (10).
3. Find the next location of the target candidate according to (11).
4. Compute $\{\hat{p}_u(\hat{\mathbf{y}}_1)\}_{u=1...m}$, and evaluate
$$\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{\mathbf{y}}_1)\hat{q}_u}\,.$$
5. While $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1), \hat{\mathbf{q}}] < \rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0), \hat{\mathbf{q}}]$
   Do $\hat{\mathbf{y}}_1 \leftarrow \frac{1}{2}(\hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1)$
   Evaluate $\rho[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1), \hat{\mathbf{q}}]$
6. If $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\| < \epsilon$ Stop.
   Otherwise Set $\hat{\mathbf{y}}_0 \leftarrow \hat{\mathbf{y}}_1$ and go to Step 2.

4.2 Implementation of the Algorithm

The stopping criterion threshold ϵ used in Step 6 is derived by constraining the vectors $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_1$ to be within the same pixel in *original* image coordinates. A lower threshold will induce subpixel accuracy. From real-time constraints (i.e., uniform CPU load in time), we also limit the number of mean shift iterations to N_{max}, typically taken equal to 20. In practice the average number of iterations is much smaller, about 4.

Implementation of the tracking algorithm can be much simpler than as presented above. The role of Step 5 is only to avoid potential numerical problems in the mean shift based maximization. These problems can appear due to the linear approximation of the Bhattacharyya coefficient. However, a large set of experiments tracking different objects for long periods of time, has shown that the Bhattacharyya coefficient computed at the new location $\hat{\mathbf{y}}_1$ failed to increase in only 0.1% of the cases. Therefore, the Step 5 is not used in practice, and as a result, there is no need to evaluate the Bhattacharyya coefficient in Steps 1 and 4.

In the practical algorithm, we only iterate by computing the weights in Step 2, deriving the new location in Step 3, and testing the size of the kernel shift in Step 6. The Bhattacharyya coefficient is computed only after the algorithm completion to evaluate the similarity between the target model and the chosen candidate.

Kernels with Epanechnikov profile [17]

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

are recommended to be used. In this case the derivative of the profile, $g(x)$, is constant and (11) reduces to

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i}{\sum_{i=1}^{n_h} w_i} \,, \tag{13}$$

i.e., a simple weighted average.

The maximization of the Bhattacharyya coefficient can be also interpreted as a matched filtering procedure. Indeed, (7) is the correlation coefficient between the unit vectors $\sqrt{\hat{\mathbf{q}}}$ and $\sqrt{\hat{\mathbf{p}}(\mathbf{y})}$, representing the target model and candidate. The mean shift procedure thus finds the local maximum of the scalar field of correlation coefficients.

Will call the *operational basin of attraction* the region in the current frame in which the new location of the target can be found by the proposed algorithm. Due to the use of kernels this basin is at least equal to the size of the target model. In other words, if in the current frame the center of the target remains in the image area covered by the target model in the previous frame, the local maximum of the Bhattacharyya coefficient is a reliable indicator for the new target location. We assume that the target representation provides sufficient discrimination, such that the Bhattacharyya coefficient presents a unique maximum in the local neighborhood.

The mean shift procedure finds a root of the gradient as function of location, which can, however, also correspond to a saddle point of the similarity surface. The saddle points are unstable solutions, and since the image noise acts as an independent perturbation factor across consecutive frames, they cannot influence the tacking performance in an image sequence.

4.3 Adaptive Scale

According to the algorithm described in Section 4.1, for a given target model, the location of the target in the current frame minimizes the distance (6) in the neighborhood of the previous location estimate. However, the scale of the target often changes in time, and thus in (4) the bandwidth h of the kernel profile has to be adapted accordingly. This is possible due to the scale invariance property of (6).

Denote by h_{prev} the bandwidth in the previous frame. We measure the bandwidth h_{opt} in the current frame by running the target localization algorithm three times, with bandwidths $h = h_{prev}$, $h = h_{prev} + \Delta h$, and $h = h_{prev} - \Delta h$. A typical value is $\Delta h = 0.1 h_{prev}$. The best result, h_{opt}, yielding the largest Bhattacharyya coefficient, is retained. To avoid over-sensitive scale adaptation, the bandwidth associated with the current frame is obtained through filtering

$$h_{new} = \gamma h_{opt} + (1 - \gamma) h_{prev} \tag{14}$$

where the default value for γ is 0.1. Note that the sequence of h_{new} contains important information about the dynamics of the target scale which can be further exploited.

4.4 Computational Complexity

Let N be the average number of iterations per frame. In Step 2 the algorithm requires to compute the representation $\{\hat{p}_u(\mathbf{y})\}_{u=1...m}$. Weighted histogram computation has roughly the same cost as the unweighted histogram since the kernel values are precomputed. In Step 3 the centroid (13) is computed which involves a weighted sum of items representing the square-root of a division of two terms. We conclude that the mean cost of the proposed algorithm for one scale is approximately given by

$$C_O = N(c_H + n_h c_S) \approx N n_h c_S \tag{15}$$

where c_H is the cost of the histogram and c_S the cost of an addition, a square-root, and a division. We assume that the number of histogram entries m and the number of target pixels n_h are in the same range.

It is of interest to compare the complexity of the new algorithm with that of target localization without gradient optimization, as discussed in [25]. The search area is assumed to be equal to the operational basin of attraction, i.e., a region covering the target model pixels. The first step is to compute n_h histograms. Assume that each histogram is derived in a squared window of n_h pixels. To implement the computation efficiently we obtain a target histogram and update it by sliding the window n_h times ($\sqrt{n_h}$ horizontal steps times $\sqrt{n_h}$ vertical steps). For each move $2\sqrt{n_h}$ additions are needed to update the histogram, hence, the effort is $c_H + 2n_h\sqrt{n_h}c_A$, where c_A is the cost of an addition. The second step is to compute n_h Bhattacharyya coefficients. This can also be done by computing one coefficient and then updating it sequentially. The effort is $mc_S + 2n_h\sqrt{n_h}c_S$. The total effort for target localization without gradient optimization is then

$$C_{NO} = c_H + 2n_h\sqrt{n_h}c_A + (m + 2n_h\sqrt{n_h})c_S \approx 2n_h\sqrt{n_h}c_S\ . \tag{16}$$

The ratio between (16) and (15) is $2 * \sqrt{n_h}/N$. In a typical setting (as it will be shown in Section 5) the target has about 50×50 pixels (i.e., $\sqrt{n_h} = 50$) and the mean number of iterations is $N = 4.19$. Thus the proposed optimization technique reduces the computational time $2 * 50/4.19 \approx 24$ times.

An optimized implementation of the new tracking algorithm has been tested on a 1GHz PC. The basic framework with scale adaptation (which involves three optimizations at each step) runs at a rate of 150 fps allowing simultaneous tracking of up to five targets in real time. Note that without scale adaptations these numbers should be multiplied by three.

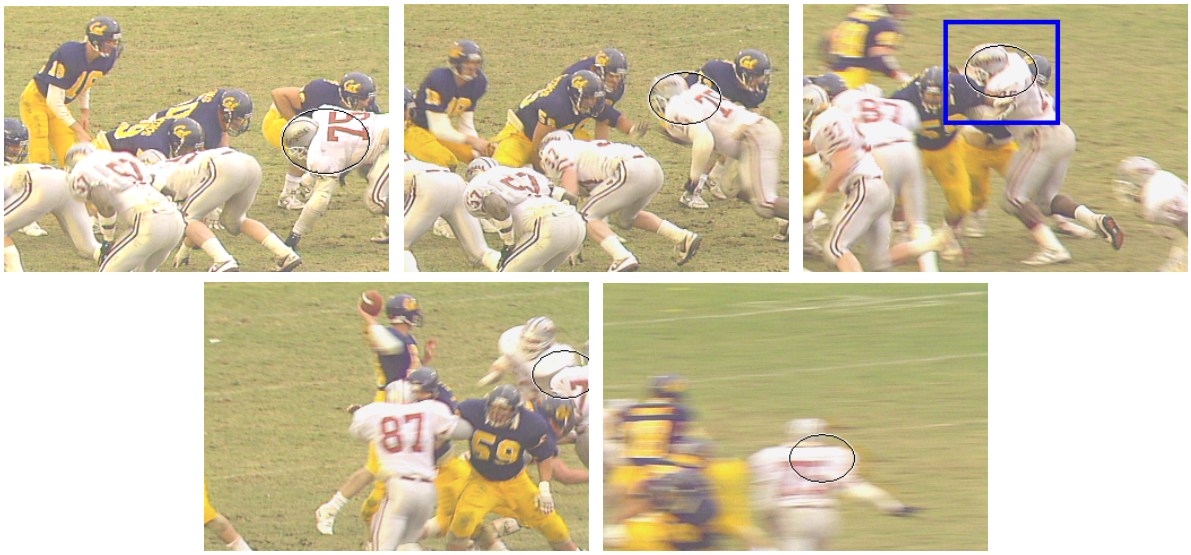


Figure 1: *Football* sequence, tracking player no. 75 . The frames 30, 75, 105, 140, and 150 are shown.

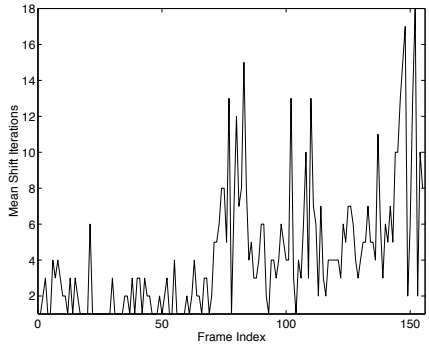


Figure 2: The number of mean shift iterations function of the frame index for the *Football* sequence. The mean number of iterations is 4.19 per frame.

5 Experimental Results

The kernel-based visual tracker was applied to many sequences and it is integrated into several applications. Here we just present some representative results. In all but the last experiments the RGB color space was taken as feature space and it was quantized into $16 \times 16 \times 16$ bins. The algorithm was implemented as discussed in Section 4.2. The Epanechnikov profile was used for histogram computations and the mean shift iterations were based on weighted averages.

The *Football* sequence (Figure 1) has 154 frames of 352×240 pixels, and the movement of player no. 75 was tracked. The target was initialized with a hand-drawn elliptical region (frame

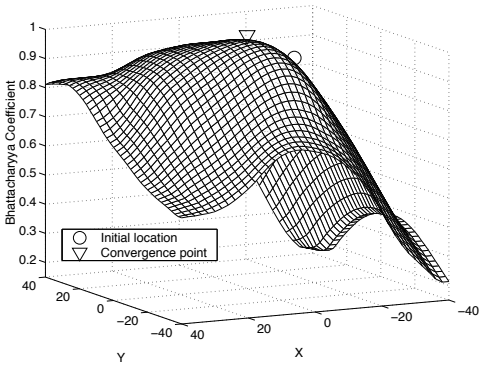


Figure 3: The similarity surface (values of the Bhattacharyya coefficient) corresponding to the rectangle marked in frame 105 of Figure 1. The initial and final locations of the mean shift iterations are also shown.

30) of size 71×53 (yielding normalization constants equal to $(h_x, h_y) = (35, 26)$). The kernel-based tracker proves to be robust to partial occlusion, clutter, distractors (frame 140) and camera motion. Since no motion model has been assumed, the tracker adapted well to the nonstationary character of the player's movements which alternates abruptly between slow and fast action. In addition, the intense blurring present in some frames due to the camera motion, does not influence the tracker performance (frame 150). The same conditions, however, can largely perturb contour based trackers.

The number of mean shift iterations necessary for each frame (one scale) is shown in Figure 2. The two central peaks correspond to the movement of the player to the center of the image and back to the left side. The last and largest peak is due to a fast move from the left to the right. In all these cases the relative large movement between two consecutive frames puts more burden on the mean shift procedure.

To demonstrate the efficiency of our approach, Figure 3 presents the surface obtained by computing the Bhattacharyya coefficient for the 81×81 pixels rectangle marked in Figure 1, frame 105. The target model (the elliptical region selected in frame 30) has been compared with the target candidates obtained by sweeping in frame 105 the elliptical region inside the rectangle. The surface is asymmetric due to neighboring colors that are similar to the target. While most of the tracking approaches based on regions [7, 27, 50] must perform an exhaustive search in the rectangle to find the maximum, our algorithm converged in four iterations as shown in Figure 3. Note that the operational basin of attraction of the mode covers the entire rectangular window.

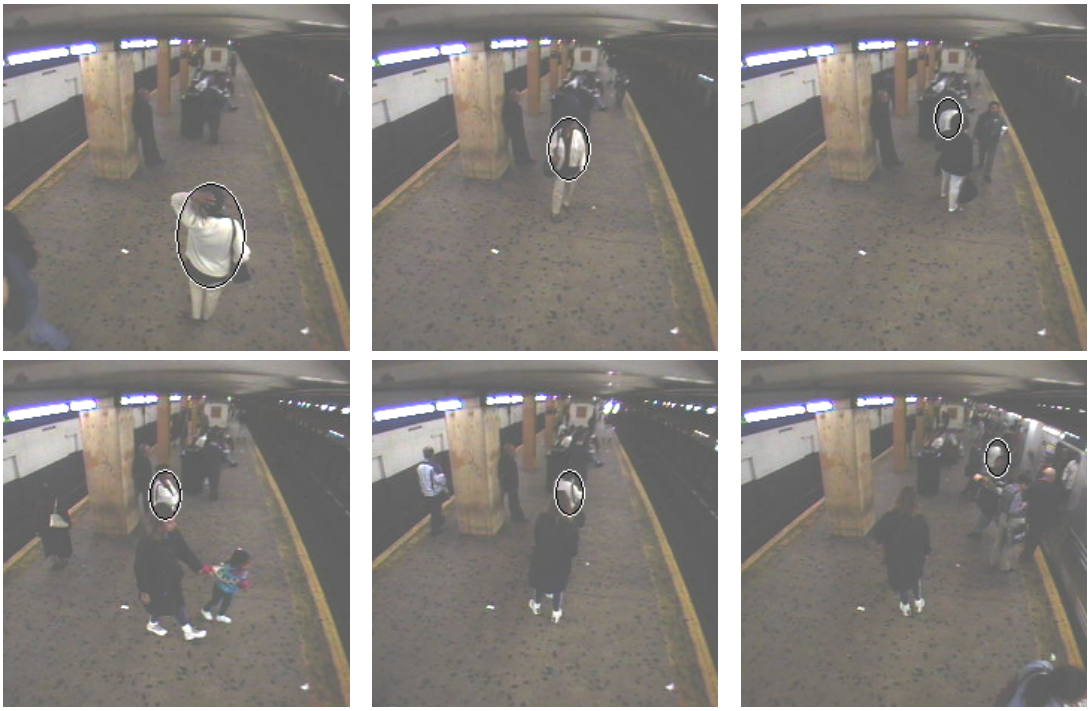


Figure 4: *Subway-1* sequence. The frames 3140, 3516, 3697, 5440, 6081, and 6681 are shown.

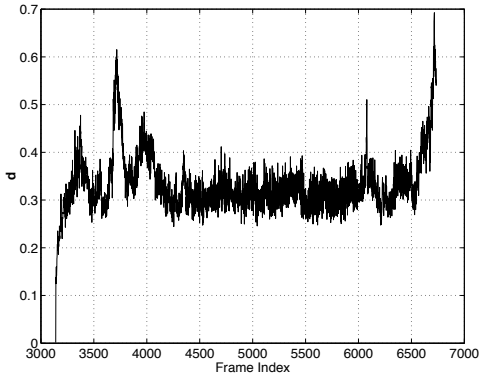


Figure 5: The minimum value of distance d function of the frame index for the *Subway-1* sequence.

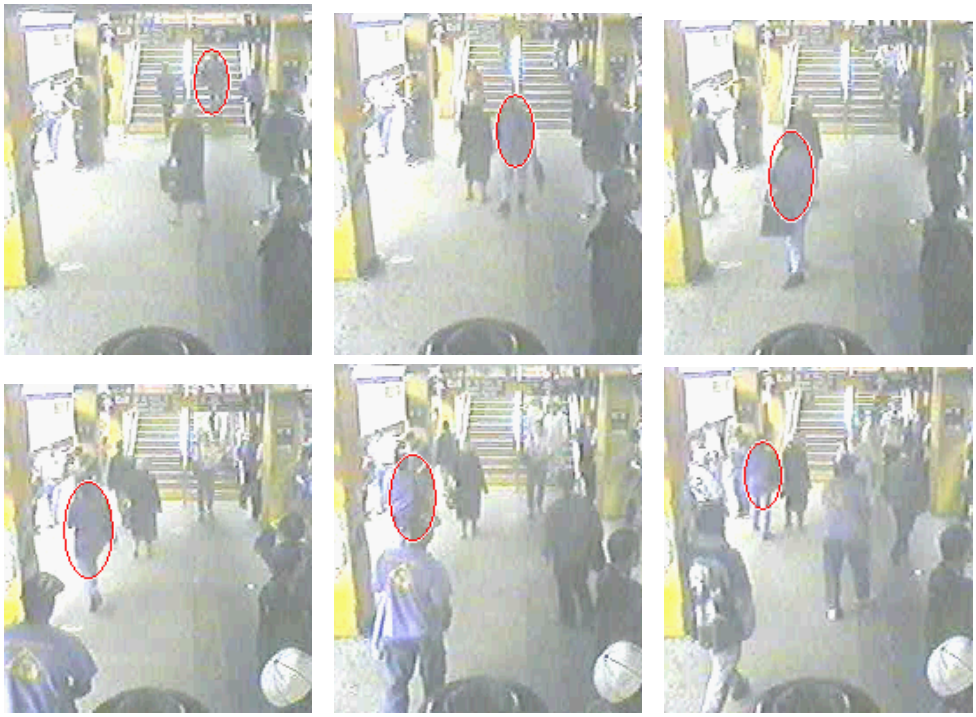


Figure 6: *Subway-2* sequence. The frames 30, 240, 330, 450, 510, and 600 are shown.

In controlled environments with fixed camera, additional geometric constraints (such as the expected scale) and background subtraction [24] can be exploited to improve the tracking process. The *Subway-1* sequence (Figure 4) is suitable for such an approach, however, the results presented here has been processed with the algorithm unchanged. This is a 2 minute sequence in which a person is tracked from the moment she enters the subway platform till she gets on the train (about 3600 frames). The tracking is made more challenging by the low quality of the sequence due to image compression artifacts. Note the changes in the size of the tracked target.

The minimum value of the distance (6) for each frame, i.e., the distance between the target model and the chosen candidate, is shown in Figure 5. The compression noise elevates the residual distance value from 0 (perfect match) to a about 0.3. Significant deviations from this value correspond to occlusions generated by other persons or rotations in depth of the target (large changes in the representation). For example, the $d \approx 0.6$ peak corresponds to the partial occlusion in frame 3697. At the end of the sequence, the person being tracked gets on the train, which produces a complete occlusion.

The shorter *Subway-2* sequence (Figure 6) of about 600 frames is even more difficult since

Figure 7: *Mug* sequence. The frames 60, 150, 240, 270, 360, and 960 are shown.

the camera quality is worse and the amount of compression is higher, introducing clearly visible artifacts. Nevertheless, the algorithm was still able to track a person through the entire sequence.

In the *Mug* sequence (Figure 7) of about 1000 frames the image of a cup (frame 60) was used as target model. The normalization constants were $(h_x = 44, h_y = 64)$. The tracker was tested for fast motion (frame 150), dramatic changes in appearance (frame 270), rotations (frame 270) and scale changes (frames 360-960).

6 Extensions of the Tracking Algorithm

We present three extensions of the basic algorithm: integration of the background information, Kalman tracking, and an application to face tracking. It should be emphasized, however, that there are many other possibilities through which the visual tracker can be further enhanced.

6.1 Background-Weighted Histogram

The background information is important for at least two reasons. First, if some of the target features are also present in the background, their relevance for the localization of the target is diminished. Second, in many applications it is difficult to exactly delineate the target, and its model might contain background features as well. At the same time, the improper use of the

background information may affect the scale selection algorithm, making impossible to measure similarity across scales, hence, to determine the appropriate target scale. Our approach is to derive a simple representation of the background features, and to use it for selecting only the salient parts from the representations of the target model and target candidates.

Let $\{\hat{o}_u\}_{u=1...m}$ (with $\sum_{u=1}^{m} \hat{o}_u = 1$) be the discrete representation (histogram) of the background in the feature space and $\hat{o}^*$ be its smallest nonzero entry. This representation is computed in a region around the target. The extent of the region is application dependent and we used an area equal to three times the target area. The weights

$$\left\{ v_u = \min\left(\frac{\hat{o}^*}{\hat{o}_u}, 1\right) \right\}_{u=1...m} \tag{17}$$

are similar in concept to the ratio histogram computed for backprojection [71]. However, in our case these weights are only employed to define a transformation for the representations of the target model and candidates. The transformation diminishes the importance of those features which have low v_u, i.e., are prominent in the background. The new target model representation is then defined by

$$\hat{q}_u = C v_u \sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2) \delta\left[b(\mathbf{x}_i^\star) - u\right] \tag{18}$$

with the normalization constant C expressed as

$$C = \frac{1}{\sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2) \sum_{u=1}^{m} v_u \delta\left[b(\mathbf{x}_i^\star) - u\right]} \,. \tag{19}$$

Compare with (2) and (3). Similarly, the new target candidate representation is

$$\hat{p}_u(\mathbf{y}) = C_h v_u \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta\left[b(\mathbf{x}_i) - u\right] \tag{20}$$

where now C_h is given by

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|^2) \sum_{u=1}^{m} v_u \delta\left[b(\mathbf{x}_i) - u\right]} \,. \tag{21}$$

An important benefit of using background-weighted histograms is shown for the *Ball* sequence (Figure 8). The movement of the ping-pong ball from frame to frame is larger than its size. Applying the technique described above, the target model can be initialized with a 21×31 size region (frame 2), larger than one obtained by accurate target delineation. The larger region yields

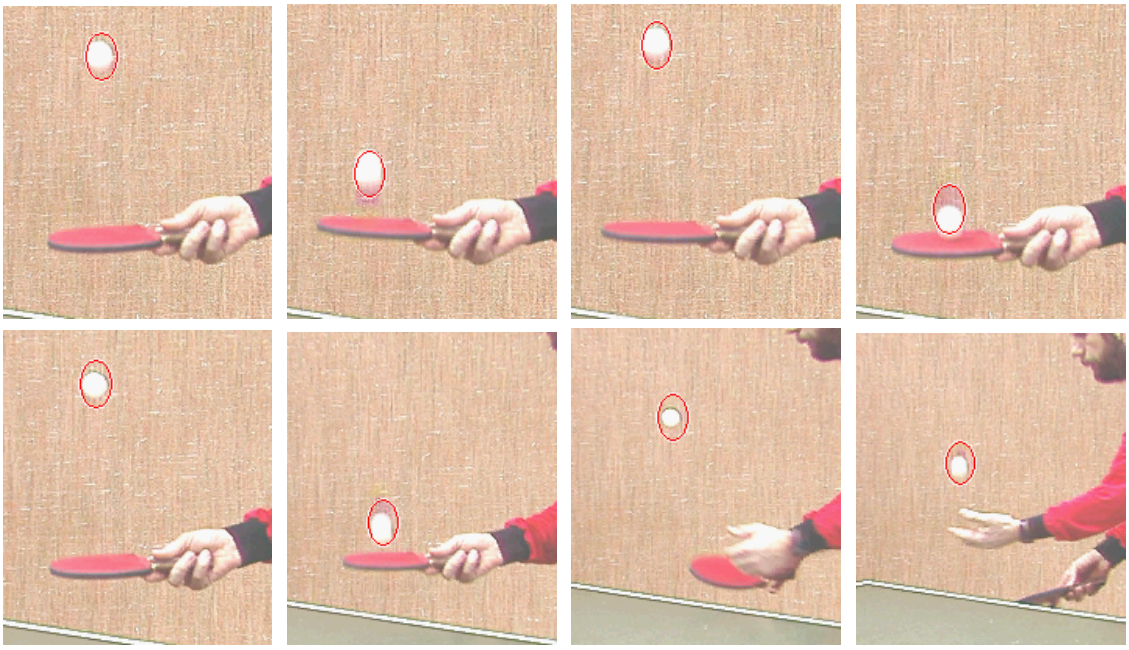


Figure 8: *Ball* sequence. The frames 2, 12, 16, 26, 32, 40, 48, and 51 are shown.

a satisfactory operational basin of attraction, while the probabilities of the colors that are part of the background are considerably reduced. The ball is reliably tracked over the entire sequence of 60 frames.

The last example is also taken from the *Football* sequence. This time the head and shoulder of player no. 59 is tracked (Figure 9). Note the changes in target appearance along the entire sequence and the rapid movements of the target.

6.2 Kalman Prediction

It was already mentioned in Section 1 that the Kalman filter assumes that the noise sequences $\mathbf{v}_k$ and $\mathbf{n}_k$ are Gaussian and the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are linear. The dynamic equation becomes $\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}$ while the measurement equation is $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k$. The matrix $\mathbf{F}$ is called the system matrix and $\mathbf{H}$ is the measurement matrix. As in the general case, the Kalman filter solves the state estimation problem in two steps: prediction and update. For more details see [5, p.56].

The kernel-based target localization method was integrated with the Kalman filtering framework. For a faster implementation, two independent trackers were defined for horizontal and vertical movement. A constant-velocity dynamic model with acceleration affected by white noise

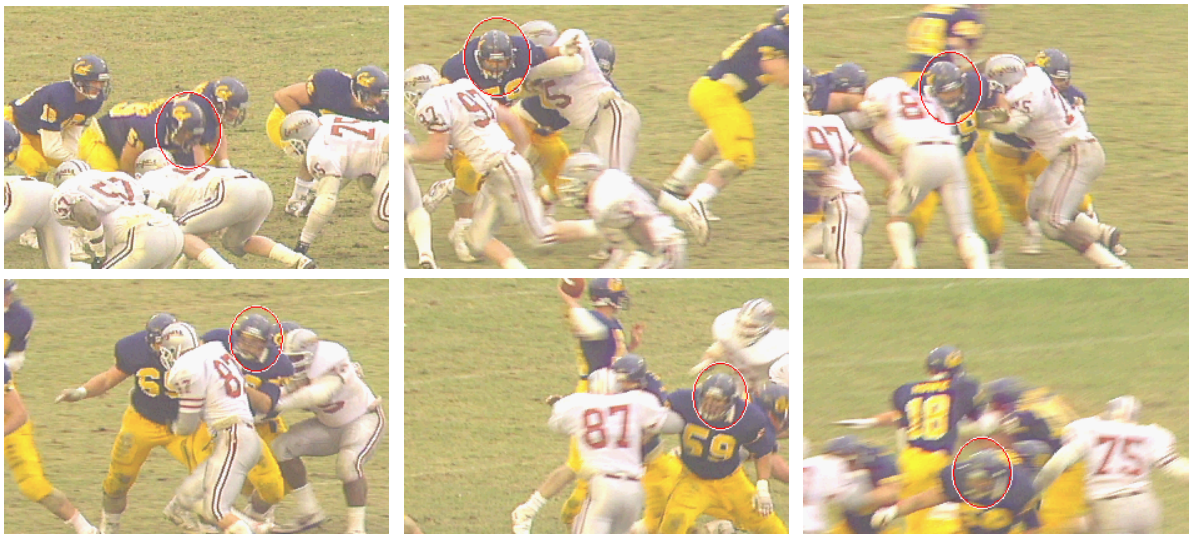


Figure 9: *Football* sequence, tracking player no. 59. The frames 70, 96, 108, 127, 140, 147 are shown.

[5, p.82] has been assumed. The uncertainty of the measurements has been estimated according to [55]. The idea is to normalize the similarity surface and represent it as a probability density function. Since the similarity surface is smooth, for each filter only 3 measurements are taken into account, one at the convergence point (peak of the surface) and the other two at a distance equal to half of the target dimension, measured from the peak. We fit a scaled Gaussian to the three points and compute the measurement uncertainty as the standard deviation of the fitted Gaussian.

A first set of tracking results incorporating the Kalman filter is presented in Figure 10 for the 120 frames *Hand* sequence where the dynamic model is assumed to be affected by a noise with standard deviation equal to 5. The size of the green cross marked on the target indicates the state uncertainty for the two trackers. Observe that the overall algorithm is able to track the target (hand) in the presence of complete occlusion by a similar object (the other hand). The presence of a similar object in the neighborhood increases the measurement uncertainty in frame 46, determining an increase in state uncertainty. In Figure 11a we present the measurements (dotted) and the estimated location of the target (continuous). Note that the graph is symmetric with respect to the number of frames since the sequence has been played forward and backward. The velocity associated with the two trackers is shown in Figure 11b.

A second set of results showing tracking with Kalman filter is displayed in Figure 12. The sequence has 187 frames of 320×240 pixels each and the initial normalization constants were

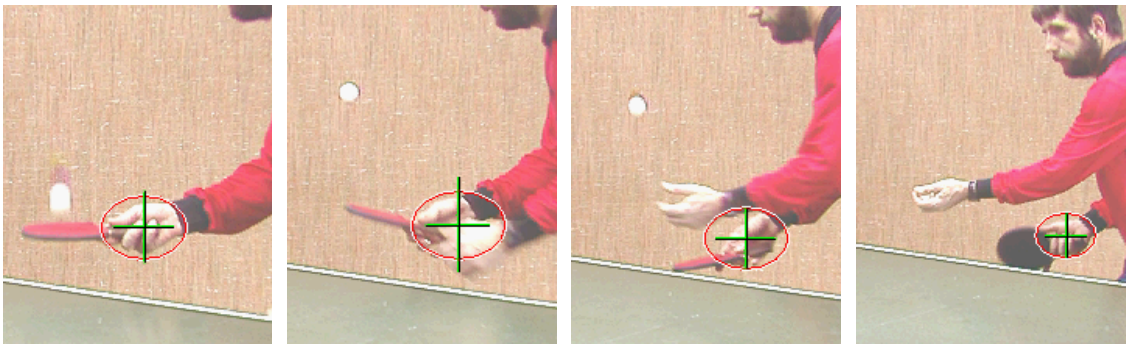
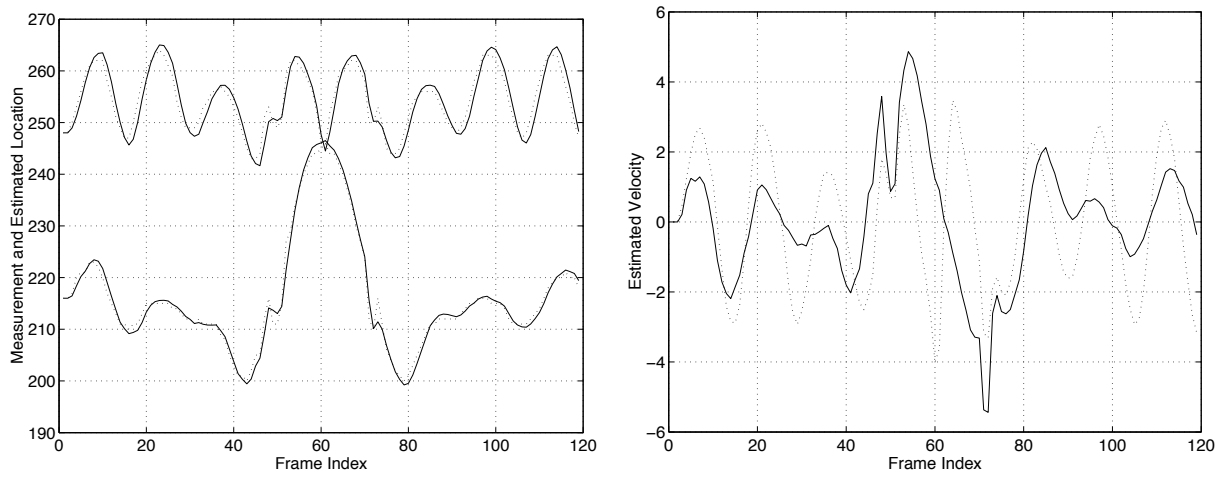


Figure 10: *Hand* sequence. The frames 40, 46, 50, and 57 are shown.



(a) (b)

Figure 11: Measurements and estimated state for *Hand* sequence. (a) The measurement value (dotted curve) and the estimated location (continuous curve) function of the frame index. Upper curves correspond to the y filter, while the lower curves correspond to the x filter. (b) Estimated velocity. Dotted curve is for the y filter and continuous curve is for the x filter

Figure 12: *Subway-3* sequence: The frames 470, 529, 580, 624, and 686 are shown.

$(h_x, h_y) = (11, 18)$. Observe the adaptation of the algorithm to scale changes. The uncertainty of the state is indicated by the green cross.

6.3 Face Tracking

We applied the proposed framework for real-time face tracking. The face model is an elliptical region whose histogram is represented in the intensity normalized rg space with 128×128 bins. To adapt to fast scale changes we also exploit the gradient information in the direction perpendicular to the border of the hypothesized face region. The scale adaptation is thus determined by maximizing the sum of two normalized scores, based on color and gradient features, respectively. In Figure 13 we present the capability of the kernel-based tracker to handle scale changes, the subject turning away (frame 150), in-plane rotations of the head (frame 498), and foreground/background saturation due to back-lighting (frame 576). The tracked face is shown in the small upper-left window.

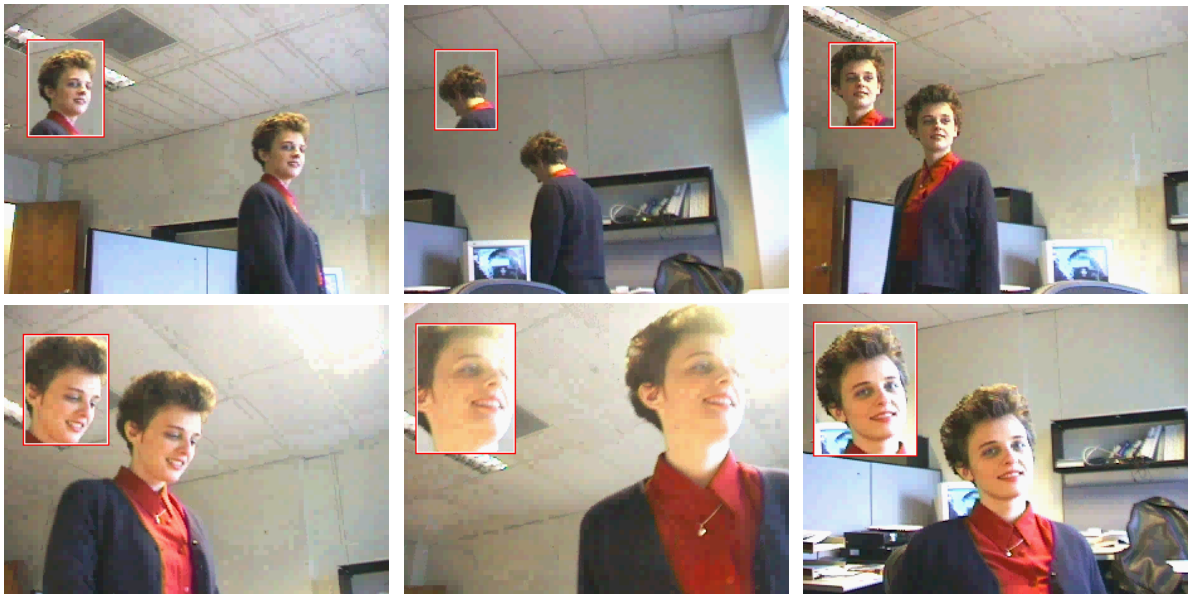


Figure 13: *Face* sequence: The frames 39, 150, 163, 498, 576, and 619 are shown.

7 Discussion

The kernel-based tracking technique introduced in this paper uses the basin of attraction of the similarity function. This function is smooth since the target representations are derived from continuous densities. Several examples validate the approach and show its efficiency. Extensions of the basic framework were presented regarding the use of background information, Kalman filtering, and face tracking. The new technique can be further combined with more sophisticated filtering and association approaches such as multiple hypothesis tracking [13].

Centroid computation has been also employed in [53]. The mean shift is used for tracking human faces by projecting the histogram of a face model onto the incoming frame [10]. However, the direct projection of the model histogram onto the new frame can introduce a large bias in the estimated location of the target, and the resulting measure is scale variant (see [37, p.262] for a discussion).

We mention that since its original publication in [18] the idea of kernel-based tracking has been exploited and developed forward by various researchers. Chen and Liu [14] experimented with the same kernel-weighted histograms, but employed the Kullback-Leibler distance as dissimilarity while performing the optimization based on trust-region methods. Haritaoglu and Flickner

[35] used an appearance model based on color and edge density in conjunction with a kernel tracker for monitoring shopping groups in stores. Yilmaz *et al*. [78] combined kernel tracking with global motion compensation for forward-looking infrared (FLIR) imagery. Xu and Fujimura [77] used night vision for pedestrian detection and tracking, where the detection is performed by a support vector machine and the tracking is kernel-based. Rao *et al*.[61] employed kernel tracking in their system for action recognition, while Caenen *et al*. [12] followed the same principle for texture analysis. The benefits of guiding random particles by gradient optimization are discussed in [70] and a particle filter for color histogram tracking based on the metric (6) is implemented in [57].

Finally we would like to add a word of caution. The tracking solution presented in this paper has several desirable properties: it is efficient, modular, has straightforward implementation, and provides superior performance on most image sequences. Nevertheless, we note that this technique should not be applied blindly. Instead, the versatility of the basic algorithm should be exploited to integrate a priori information which is almost always available when a specific application is considered. For example, if the motion of the target from frame to frame is known to be larger than the operational basin of attraction, one should initialize the tracker in multiple locations in the neighborhood of basin of attraction, according to the motion model. If occlusions are present, one should employ a more sophisticated motion filter. Similarly, one should verify that the chosen target representation is sufficiently discriminant for the application domain. The kernel-based tracking technique, when combined with prior task-specific information, can achieve reliable performance. [2]

APPENDIX

Proof that the distance $d(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \sqrt{1 - \rho(\hat{\mathbf{p}}, \hat{\mathbf{q}})}$ is a metric

The proof is based on the properties of the Bhattacharyya coefficient (7). According to the Jensen's inequality [19, p.25] we have

$$\rho(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u \hat{q}_u} = \sum_{u=1}^{m} \hat{p}_u \sqrt{\frac{\hat{q}_u}{\hat{p}_u}} \leq \sqrt{\sum_{u=1}^{m} \hat{q}_u} = 1, \tag{A.1}$$

[2] A patent application has been filed covering the tracking algorithm together with the extensions and various applications [79].

with equality iff $\hat{\mathbf{p}} = \hat{\mathbf{q}}$. Therefore, $d(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \sqrt{1 - \rho(\hat{\mathbf{p}}, \hat{\mathbf{q}})}$ exists for all discrete distributions $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$, is positive, symmetric, and is equal to zero iff $\hat{\mathbf{p}} = \hat{\mathbf{q}}$.

To prove the triangle inequality consider three discrete distributions defining the m-dimensional vectors $\hat{\mathbf{p}}$, $\hat{\mathbf{q}}$, and $\hat{\mathbf{r}}$, associated with the points $\boldsymbol{\xi}_p = \left(\sqrt{\hat{p}_1}, \ldots, \sqrt{\hat{p}_m}\right)^\top$, $\boldsymbol{\xi}_q = \left(\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m}\right)^\top$, and $\boldsymbol{\xi}_r = \left(\sqrt{\hat{r}_1}, \ldots, \sqrt{\hat{r}_m}\right)^\top$ on the unit hypersphere. From the geometric interpretation of the Bhattacharyya coefficient, the triangle inequality

$$d(\hat{\mathbf{p}}, \hat{\mathbf{r}}) + d(\hat{\mathbf{q}}, \hat{\mathbf{r}}) \geq d(\hat{\mathbf{p}}, \hat{\mathbf{q}}) \tag{A.2}$$

is equivalent to

$$\sqrt{1 - cos(\boldsymbol{\xi}_p, \boldsymbol{\xi}_r)} + \sqrt{1 - cos(\boldsymbol{\xi}_q, \boldsymbol{\xi}_r)} \geq \sqrt{1 - cos(\boldsymbol{\xi}_p, \boldsymbol{\xi}_q)}\,. \tag{A.3}$$

If we fix the points $\boldsymbol{\xi}_p$ and $\boldsymbol{\xi}_q$, and the angle between $\boldsymbol{\xi}_p$ and $\boldsymbol{\xi}_r$, the left side of inequality (A.3) is minimized when the vectors $\boldsymbol{\xi}_p, \boldsymbol{\xi}_q$, and $\boldsymbol{\xi}_r$ lie in the same plane. Thus, the inequality (A.3) can be reduced to a 2-dimensional problem that can be easily proven by employing the half-angle sinus formula and a few trigonometric manipulations.

Acknowledgment

A preliminary version of the paper was presented at the *2000 IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC. Peter Meer was supported by the National Science Foundation under the award IRI 99-87695.

References

[1] J. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer Vision and Image Understanding*, vol. 73, pp. 428–440, 1999.

[2] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–189, 2002.

[4] S. Avidan, "Support vector tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume I, 2001, pp. 184–191.

[5] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

[6] B. Bascle and R. Deriche, "Region tracking through image sequences," in *Proc. 5th Intl. Conf. on Computer Vision,* Cambridge, MA, 1995, pp. 302–307.

[7] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Santa Barbara, CA, 1998, pp. 232–237.

[8] M. Black and D. Fleet, "Probabilistic detection and tracking of motion boundaries," *Intl. J. of Computer Vision*, vol. 38, no. 3, pp. 231–245, 2000.

[9] Y. Boykov and D. Huttenlocher, "Adaptive Bayesian recognition in tracking rigid objects," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, 2000, pp. 697–704.

[10] G. R. Bradski, "Computer vision face tracking as a component of a perceptual user interface," in *Proc. IEEE Workshop on Applications of Computer Vision,* Princeton, NJ, October 1998, pp. 214–219.

[11] A. D. Bue, D. Comaniciu, V. Ramesh, and C. Regazzoni, "Smart cameras with real-time video object generation," in *Proc. IEEE Intl. Conf. on Image Processing,* Rochester, NY, volume III, 2002, pp. 429–432.

[12] G. Caenen, V. Ferrari, A. Zalesny, and L. VanGool, "Analyzing the layout of composite textures," in *Proceedings Texture 2002 Workshop,* Copenhagen, Denmark, 2002, pp. 15–19.

[13] T. Cham and J. Rehg, "A multiple hypothesis approach to figure tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, volume II, 1999, pp. 239–219.

[14] H. Chen and T. Liu, "Trust-region methods for real-time tracking," in *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, volume II, 2001, pp. 717–722.

[15] Y. Chen, Y. Rui, and T. Huang, "JPDAF-based HMM for real-time contour tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume I, 2001, pp. 543–550.

[16] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001.

[17] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[18] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, volume II, June 2000, pp. 142–149.

[19] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[20] I. Cox and S. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 2, pp. 138–150, 1996.

[21] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *Intl. J. of Computer Vision*, vol. 38, no. 2, pp. 99–127, 2000.

[22] A. Djouadi, O. Snorrason, and F. Garber, "The quality of training-sample estimates of the Bhattacharyya coefficient," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 92–97, 1990.

[23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[24] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. European Conf. on Computer Vision,* Dublin, Ireland, volume II, June 2000, pp. 751–767.

[25] F. Ennesser and G. Medioni, "Finding Waldo, or focus of attention using local color information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 805–809, 1995.

[26] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Real-time affine region tracking and coplanar grouping," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume II, 2001, pp. 226–233.

[27] P. Fieguth and D. Teropoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, 1997, pp. 21–27.

[28] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, second edition, 1990.

[29] J. Garcia, J. Valdivia, and X. Vidal, "Information theoretic measure for visual target distinctness," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 4, pp. 362–383, 2001.

[30] D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, pp. 82–98, 1999.

[31] N. Gordon, D. Salmond, and A. Smith, "A novel approach to non-linear and non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, pp. 107–113, 1993.

[32] M. Greiffenhagen, D. Comaniciu, H. Niemann, and V. Ramesh, "Design, analysis and engineering of video monitoring systems: An approach and a case study," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1498–1517, 2001.

[33] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Francisco, CA, 1996, pp. 403–410.

[34] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen, "Computer vision for driver assistance systems," in *Proceedings SPIE*, volume 3364, 1998, pp. 136–147.

[35] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, 2001.

[36] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who? When? Where? What? - A real time system for detecting and tracking people," in *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition,* Nara, Japan, 1998, pp. 222–227.

[37] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Spatial color indexing and applications," *Intl. J. of Computer Vision*, vol. 35, no. 3, pp. 245–268, 1999.

[38] C. Hue, J. Cadre, and P. Perez, "Sequential Monte Carlo filtering for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, 2002.

[39] S. Intille, J. Davis, and A. Bobick, "Real-time closed-world tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, 1997, pp. 697–703.

[40] M. Isard and A. Blake, "Condensation - Conditional density propagation for visual tracking," *Intl. J. of Computer Vision*, vol. 29, no. 1, 1998.

[41] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hawaii, volume I, 2001, pp. 415–422.

[42] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. SPIE*, volume 3068, 1997, pp. 182–193.

[43] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Tech.*, vol. 15, pp. 52–60, 1967.

[44] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, 1999, pp. 253–259.

[45] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," *J. of Amer. Stat. Assoc.*, vol. 82, pp. 1032–1063, 1987.

[46] S. Konishi, A. Yuille, J. Coughlan, and S. Zhu, "Fundamental bounds on edge detection: An information theoretic evaluation of different edge cues," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, 1999, pp. 573–579.

[47] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for EasyLiving," in *Proc. IEEE Intl. Workshop on Visual Surveillance,* Dublin, Ireland, 2000, pp. 3–10.

[48] B. Li and R. Chellappa, "Simultaneous tracking and verification via sequential posterior estimation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, volume II, 2000, pp. 110–117.

[49] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Information Theory*, vol. 37, pp. 145–151, 1991.

[50] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in *IEEE Workshop on Applications of Computer Vision,* Princeton, NJ, 1998, pp. 8–14.

[51] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Intl. J. of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[52] R. Mahler, "Engineering statistics for multi-object tracking," in *Proc. IEEE Workshop Multi-Object Tracking*, 2001.

[53] S. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixture models," *Image and Vision Computing Journal*, vol. 17, pp. 223–229, 1999.

[54] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[55] K. Nickels and S. Hutchinson, "Estimating uncertainty in SSD-based feature tracking," *Image and Vision Computing*, vol. 20, pp. 47–58, 2002.

[56] C. Olson, "Image registration by aligning entropies," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume II, 2001, pp. 331–336.

[57] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. European Conf. on Computer Vision,* Copenhagen, Denmark, volume I, 2002, pp. 661–675.

[58] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[59] J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in *Proc. 7th Intl. Conf. on Computer Vision,* Kerkyra, Greece, 1999, pp. 1165–1173.

[60] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[61] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *Intl. J. of Computer Vision*, vol. 50, no. 2, p. To appear, 2003.

[62] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 6, pp. 560–576, 2001.

[63] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control*, vol. AC-24, pp. 843–854, 1979.

[64] A. Roche, G. Malandain, and N. Ayache, "Unifying maximum likelihood approaches in medical image registration," Technical Report 3741, INRIA, 1999.

[65] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, 1999, pp. 117–123.

[66] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume II, 2001, pp. 786–793.

[67] S. Sclaroff and J. Isidoro, "Active blobs," in *Proc. 6th Intl. Conf. on Computer Vision,* Bombay, India, 1998, pp. 1146–1153.

[68] D. W. Scott, *Multivariate Density Estimation.* Wiley, 1992.

[69] C. Sminchisescu and B. Triggs, "Covariance scaled sampling for monocular 3D body tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Kauai, Hawaii, volume I, 2001, pp. 447–454.

[70] J. Sullivan and J. Rittscher, "Guiding random particles by deterministic search," in *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, volume I, 2001, pp. 323–330.

[71] M. Swain and D. Ballard, "Color indexing," *Intl. J. of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[72] P. Viola and W. Wells, "Alignment by maximization of mutual information," *Intl. J. of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[73] S. Wachter and H. Nagel, "Tracking persons in monocular image sequences," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 174–192, 1999.

[74] R. Wildes, R. Kumar, H. Sawhney, S. Samasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, D. Hirvonen, M. Hansen, and P. Burt, "Aerial video surveillance and exploitation," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1518–1539, 2001.

[75] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, 1997.

[76] Y. Wu and T. Huang, "A co-inference approach to robust tracking," in *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, volume II, 2001, pp. 26–33.

[77] F. Xu and K. Fujimura, "Pedestrian detection and tracking with night vision," in *Proc. IEEE Intelligent Vehicle Symposium,* Versailles, France, 2002.

[78] A. Yilmaz, K. Shafique, N. Lobo, X. Li, T. Olson, and M. Shah, "Target tracking in FLIR imagery using mean shift and global motion compensation," in *IEEE Workshop on Computer Vision Beyond Visible Spectrum,* Kauai, Hawaii, 2001.

[79] "Real-time tracking of non-rigid objects using mean shift." US patent pending, 2000.