Visual Tracking

- Comaniciu, V. Ramesh and P. Meer, Real-time Tracking of Non-Rigid Objects using Mean Shift, (Hilton Head Island, South Carolina, USA), pp. 42- 151, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,2000
- Nummiaro, K. and Koller-Meier, E. and Van Gool, L., An adaptive color-based particle filter, *Image and Vision Computing*, 21(1), pp.99-110, 2003.

Real-Time Tracking of Non-Rigid Objects using Mean Shift

Dorin Comaniciu Visvanathan Ramesh
Imaging & Visualization Department
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540

Peter Meer
Electrical & Computer Engineering Department
Rutgers University
94 Brett Road, Piscataway, NJ 08855

Abstract

A new method for real-time tracking of non-rigid objects seen from a moving camera is proposed. The central computational module is based on the mean shift iterations and finds the most probable target position in the current frame. The dissimilarity between the target model (its color distribution) and the target candidates is expressed by a metric derived from the Bhattacharyya coefficient. The theoretical analysis of the approach shows that it relates to the Bayesian framework while providing a practical, fast and efficient solution. The capability of the tracker to handle in real-time partial occlusions, significant clutter, and target scale variations, is demonstrated for several image sequences.

1 Introduction

The efficient tracking of visual features in complex environments is a challenging task for the vision community. Real-time applications such as surveillance and monitoring [10], perceptual user interfaces [4], smart rooms [16, 28], and video compression [12] all require the ability to track moving objects. The computational complexity of the tracker is critical for most applications, only a small percentage of a system resources being allocated for tracking, while the rest is assigned to preprocessing stages or to high-level tasks such as recognition, trajectory interpretation, and reasoning [24].

This paper presents a new approach to the real-time tracking of non-rigid objects based on visual features such as color and/or texture, whose statistical distributions characterize the object of interest. The proposed tracking is appropriate for a large variety of objects with different color/texture patterns, being robust to partial occlusions, clutter, rotation in depth, and changes in camera position. It is a natural application to motion analysis of the mean shift procedure introduced earlier [6, 7]. The mean shift iterations are employed to find the target candidate that is the most similar to a given target model, with the similarity being expressed by a metric based on the Bhattacharyya coefficient. Various test sequences showed the superior tracking performance, obtained with low computational complexity.

The paper is organized as follows. Section 2 presents and extends the mean shift property. Section 3 introduces the metric derived from the Bhattacharyya coefficient. The tracking algorithm is developed and analyzed in Section 4. Experiments and comparisons are given in Section 5, and the discussions are in Section 6.

2 Mean Shift Analysis

We define next the sample mean shift, introduce the iterative mean shift procedure, and present a new theorem showing the convergence for kernels with convex and monotonic profiles. For applications of the mean shift property in low level vision (filtering, segmentation) see [6].

2.1 Sample Mean Shift

Given a set $\{\mathbf{x}_i\}_{i=1\ldots n}$ of n points in the d-dimensional space R^d, the *multivariate kernel density estimate* with kernel $K(\mathbf{x})$ and window radius (bandwidth) h, computed in the point $\mathbf{x}$ is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right). \qquad (1)$$

The minimization of the average global error between the estimate and the true density yields the multivariate Epanechnikov kernel [25, p.139]

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\|<1 \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where c_d is the volume of the unit d-dimensional sphere. Another commonly used kernel is the multivariate normal

$$K_N(\mathbf{x}) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right). \qquad (3)$$

Let us introduce the *profile* of a kernel K as a function $k : [0,\infty) \rightarrow R$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$. For example, according to (2) the Epanechnikov profile is

$$k_E(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-x) & \text{if } x<1 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

and from (3) the normal profile is given by

$$k_N(x) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}x\right). \qquad (5)$$

Employing the profile notation we can write the density estimate (1) as

$$\hat{f}_K(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \qquad (6)$$

We denote

$$g(x) = -k'(x)\ , \qquad (7)$$

assuming that the derivative of k exists for all $x \in [0,\infty)$, except for a finite set of points. A kernel G can be defined as

$$G(\mathbf{x}) = Cg(\|\mathbf{x}\|^2), \qquad (8)$$

where C is a normalization constant. Then, by taking the estimate of the density gradient as the gradient of the density estimate we have

$$\hat{\nabla} f_K(\mathbf{x}) \equiv \nabla \hat{f}_K(\mathbf{x}) = \frac{2}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}-\mathbf{x}_i)\, k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

$$= \frac{2}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i-\mathbf{x})\, g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) = \frac{2}{nh^{d+2}} \times$$

$$\times\left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right], \quad (9)$$

where $\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$ can be assumed to be nonzero. Note that the derivative of the Epanechnikov profile is the uniform profile, while the derivative of the normal profile remains a normal.

The last bracket in (9) contains the sample mean shift vector

$$M_{h,G}(\mathbf{x}) \equiv \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (10)$$

and the density estimate at $\mathbf{x}$

$$\hat{f}_G(\mathbf{x}) \equiv \frac{C}{nh^d} \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \quad (11)$$

computed with kernel G. Using now (10) and (11), (9) becomes

$$\hat{\nabla} f_K(\mathbf{x}) = \hat{f}_G(\mathbf{x}) \frac{2/C}{h^2} M_{h,G}(\mathbf{x}) \quad (12)$$

from where it follows that

$$M_{h,G}(\mathbf{x}) = \frac{h^2}{2/C} \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})}. \quad (13)$$

Expression (13) shows that the sample mean shift vector obtained with kernel G is an estimate of the normalized density gradient obtained with kernel K. This is a more general formulation of the property first remarked by Fukunaga [15, p. 535].

2.2 A Sufficient Convergence Condition

The *mean shift procedure* is defined recursively by computing the mean shift vector $M_{h,G}(\mathbf{x})$ and translating the center of kernel G by $M_{h,G}(\mathbf{x})$.

Let us denote by $\left\{\mathbf{y}_j\right\}_{j=1,2\ldots}$ the sequence of successive locations of the kernel G, where

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_j-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{y}_j-\mathbf{x}_i}{h}\right\|^2\right)}, \quad j = 1, 2, \ldots \quad (14)$$

is the weighted mean at $\mathbf{y}_j$ computed with kernel G and $\mathbf{y}_1$ is the center of the initial kernel. The density estimates computed with kernel K in the points (14) are

$$\hat{f}_K = \left\{\hat{f}_K(j)\right\}_{j=1,2\ldots} \equiv \left\{\hat{f}_K(\mathbf{y}_j)\right\}_{j=1,2\ldots}. \quad (15)$$

These densities are only implicitly defined to obtain $\hat{\nabla} f_K$. However we need them to prove the convergence of the sequences (14) and (15).

Theorem 1 *If the kernel K has a convex and monotonic decreasing profile and the kernel G is defined according to (7) and (8), the sequences (14) and (15) are convergent.*

The Theorem 1 generalizes the convergence shown in [6], where K was the Epanechnikov kernel, and G the uniform kernel. Its proof is given in the Appendix. Note that Theorem 1 is also valid when we associate to each data point $\mathbf{x}_i$ a positive weight w_i.

3 Bhattacharyya Coefficient Based Metric for Target Localization

The task of finding the target location in the current frame is formulated as follows. The feature $\mathbf{z}$ representing the color and/or texture of the target model is assumed to have a density function $q_{\mathbf{z}}$, while the target candidate centered at location $\mathbf{y}$ has the feature distributed according to $p_{\mathbf{z}}(\mathbf{y})$. The problem is then to find the discrete location $\mathbf{y}$ whose associated density $p_{\mathbf{z}}(\mathbf{y})$ is the most similar to the target density $q_{\mathbf{z}}$.

To define the similarity measure we take into account that the probability of classification error in statistical hypothesis testing is directly related to the similarity of the two distributions. The larger the probability of error, the more similar the distributions. Therefore, (contrary to the hypothesis testing), we formulate the target location estimation problem as the derivation of the estimate that *maximizes* the Bayes error associated with the model and candidate distributions. For the moment, we assume that the target has equal prior probability to be present at any location $\mathbf{y}$ in the neighborhood of the previously estimated location.

An entity closely related to the Bayes error is the Bhattacharyya coefficient, whose general form is defined by [19]

$$\rho(\mathbf{y}) \equiv \rho\left[p(\mathbf{y}), q\right] = \int \sqrt{p_{\mathbf{z}}(\mathbf{y}) q_{\mathbf{z}}}\, d\mathbf{z}\ . \quad (16)$$

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 [11, 19].

Our interest in expression (16) is, however, motivated by its near optimality given by the relationship to the Bayes error. Indeed, let us denote by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ two sets of parameters for the distributions p and q and by $\boldsymbol{\pi} = (\pi_p, \pi_q)$ a set of prior probabilities. If the value of (16) is smaller for the set $\boldsymbol{\alpha}$ than for the set $\boldsymbol{\beta}$, it can be proved [19] that, there exists a set of priors $\boldsymbol{\pi}^*$ for which the error probability for the set $\boldsymbol{\alpha}$ is less than the error probability for the set $\boldsymbol{\beta}$. In addition, starting from (16) upper and lower error bounds can be derived for the probability of error.

The derivation of the Bhattacharyya coefficient from sample data involves the estimation of the densities p and q, for which we employ the histogram formulation. Although not the best nonparametric density estimate [25], the histogram satisfies the low computational cost imposed by real-time processing. We estimate the discrete density $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\dots m}$ (with $\sum_{u=1}^{m}\hat{q}_u = 1$) from the m-bin histogram of the target model, while $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\dots m}$ (with $\sum_{u=1}^{m}\hat{p}_u = 1$) is estimated at a given location $\mathbf{y}$ from the m-bin histogram of the target candidate. Hence, the sample estimate of the Bhattacharyya coefficient is given by

$$\hat{\rho}(\mathbf{y}) \equiv \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u}. \qquad (17)$$

The geometric interpretation of (17) is the cosine of the angle between the m-dimensional, unit vectors $\left(\sqrt{\hat{p}_1}, \dots, \sqrt{\hat{p}_m}\right)^\top$ and $\left(\sqrt{\hat{q}_1}, \dots, \sqrt{\hat{q}_m}\right)^\top$.

Using now (17) the distance between two distributions can be defined as

$$d(\mathbf{y}) = \sqrt{1 - \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right]}\ . \qquad (18)$$

The statistical measure (18) is well suited for the task of target localization since:

1. It is nearly optimal, due to its link to the Bayes error. Note that the widely used histogram intersection technique [26] has no such theoretical foundation.
2. It imposes a metric structure (see Appendix). The Bhattacharyya distance [15, p.99] or Kullback divergence [8, p.18] are not metrics since they violate at least one of the distance axioms.
3. Using discrete densities, (18) is invariant to the scale of the target (up to quantization effects). Histogram intersection is scale variant [26].
4. Being valid for arbitrary distributions, the distance (18) is superior to the Fisher linear discriminant, which yields useful results only for distributions that are separated by the mean-difference [15, p.132].

Similar measures were already used in computer vision. The Chernoff and Bhattacharyya bounds have been employed in [20] to determine the effectiveness of edge detectors. The Kullback divergence has been used in [27] for finding the pose of an object in an image.

The next section shows how to minimize (18) as a function of $\mathbf{y}$ in the neighborhood of a given location, by exploiting the mean shift iterations. Only the distribution of the object colors will be considered, although the texture distribution can be integrated into the same framework.

4 Tracking Algorithm

We assume in the sequel the support of two modules which should provide (a) detection and localization in the initial frame of the objects to track (targets) [21, 23], and (b) periodic analysis of each object to account for possible updates of the target models due to significant changes in color [22].

4.1 Color Representation

Target Model Let $\{\mathbf{x}_i^\star\}_{i=1\dots n}$ be the pixel locations of the target model, centered at $\mathbf{0}$. We define a function $b : R^2 \rightarrow \{1 \dots m\}$ which associates to the pixel at location $\mathbf{x}_i^\star$ the index $b(\mathbf{x}_i^\star)$ of the histogram bin corresponding to the color of that pixel. The probability of the color u in the target model is derived by employing a convex and monotonic decreasing kernel profile k which assigns a smaller weight to the locations that are farther from the center of the target. The weighting increases the robustness of the estimation, since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or background. The radius of the kernel profile is taken equal to one, by assuming that the generic coordinates x and y are normalized with h_x and h_y, respectively. Hence, we can write

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2)\delta\left[b(\mathbf{x}_i^\star) - u\right]\ , \qquad (19)$$

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

$$C = \frac{1}{\sum_{i=1}^{n} k(\|\mathbf{x}_i^\star\|^2)}\ , \qquad (20)$$

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

Target Candidates Let $\{\mathbf{x}_i\}_{i=1\dots n_h}$ be the pixel locations of the target candidate, centered at $\mathbf{y}$ in the current frame. Using the same kernel profile k, but with radius h, the probability of the color u in the target candidate is given by

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

where C_h is the normalization constant. The radius of the kernel profile determines the number of pixels (i.e., the scale) of the target candidate. By imposing the condition that $\sum_{u=1}^{m}\hat{p}_u = 1$ we obtain

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\|^2)}. \qquad (22)$$

Note that C_h does not depend on $\mathbf{y}$, since the pixel locations $\mathbf{x}_i$ are organized in a regular lattice, $\mathbf{y}$ being one of the lattice nodes. Therefore, C_h can be precalculated for a given kernel and different values of h.

4.2 Distance Minimization

According to Section 3, the most probable location $\mathbf{y}$ of the target in the current frame is obtained by minimizing the distance (18), which 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 estimated location $\hat{\mathbf{y}}_0$ of the target in the previous frame. Thus, the color 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 Bhattacharyya coefficient (17) is approximated as (after some manipulations)

$$\rho\left[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}\right] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\mathbf{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \qquad (23)$$

where it is assumed that the target candidate $\{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}$ does not change drastically from the initial $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$, and that $\hat{p}_u(\hat{\mathbf{y}}_0) > 0$ for all $u = 1\ldots m$. Introducing now (21) in (23) we obtain

$$\rho\left[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}\right] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right) \qquad (24)$$

where

$$w_i = \sum_{u=1}^{m}\delta\left[b(\mathbf{x}_i)-u\right]\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}. \qquad (25)$$

Thus, to minimize the distance (18), the second term in equation (24) has to be maximized, the first term being independent of $\mathbf{y}$. The second term represents the density estimate computed with kernel profile k at $\mathbf{y}$ in the current frame, with the data being weighted by w_i (25). The maximization can be efficiently achieved based on the mean shift iterations, using the following algorithm.

Bhattacharyya Coefficient $\rho\left[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}\right]$ *Maximization*
Given the distribution $\{\hat{q}_u\}_{u=1\ldots m}$ of the target model and the estimated location $\hat{\mathbf{y}}_0$ of the target in the previous frame:

1. Initialize the location of the target in the current frame with $\hat{\mathbf{y}}_0$, compute the distribution $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$, and evaluate
$$\rho\left[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0),\hat{\mathbf{q}}\right] = \textstyle\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathbf{y}_0)\hat{q}_u}\ .$$

2. Derive the weights $\{w_i\}_{i=1\ldots n_h}$ according to (25).

3. Based on the mean shift vector, derive the new location of the target (14)
$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h}\mathbf{x}_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0-\mathbf{x}_i}{h}\right\|^2\right)}\ . \qquad (26)$$
Update $\{\hat{p}_u(\hat{\mathbf{y}}_1)\}_{u=1\ldots m}$, and evaluate
$$\rho\left[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1),\hat{\mathbf{q}}\right] = \textstyle\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathbf{y}_1)\hat{q}_u}\ .$$

4. While $\rho\left[\hat{\mathbf{p}}(\hat{\mathbf{y}}_1),\hat{\mathbf{q}}\right] < \rho\left[\hat{\mathbf{p}}(\hat{\mathbf{y}}_0),\hat{\mathbf{q}}\right]$
   Do $\hat{\mathbf{y}}_1 \leftarrow \frac{1}{2}(\hat{\mathbf{y}}_0+\hat{\mathbf{y}}_1)$.

5. 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 1.

The proposed optimization employs the mean shift vector in Step 3 to increase the value of the approximated Bhattacharyya coefficient expressed by (24). Since this operation does not necessarily increase the value of $\rho\left[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}\right]$, the test included in Step 4 is needed to validate the new location of the target. However, practical experiments (tracking different objects, for long periods of time) showed that the Bhattacharyya coefficient computed at the location defined by equation (26) was almost always larger than the coefficient corresponding to $\hat{\mathbf{y}}_0$. Less than 0.1% of the performed maximizations yielded cases where the Step 4 iterations were necessary. The termination threshold ϵ used in Step 5 is derived by constraining the vectors representing $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_1$ to be within the same pixel in image coordinates.

The tracking consists in running for each frame the optimization algorithm described above. Thus, given the target model, the new location of the target in the current frame minimizes the distance (18) in the neighborhood of the previous location estimate.

4.3 Scale Adaptation

The scale adaptation scheme exploits the property of the distance (18) to be invariant to changes in the object scale. We simply modify the radius h of the kernel profile with a certain fraction (we used ±10%), let the tracking algorithm to converge again, and choose the radius yielding the largest decrease in the distance (18). An IIR filter is used to derive the new radius based on the current measurements and old radius.

5 Experiments

The proposed method has been applied to the task of tracking a football player marked by a hand-drawn ellipsoidal region (first image of Figure 1). The sequence has 154 frames of 352×240 pixels each and the initial normalization constants (determined from the size of the target model) were $(h_x, h_y) = (71, 53)$. The Epanechnikov profile (4) has been used for histogram computation, therefore, the mean shift iterations were computed with the uniform profile. The target histogram has been derived in the RGB space with $32 \times 32 \times 32$ bins. The algorithm runs comfortably at 30 fps on a 600 MHz PC, Java implementation.

The tracking results are presented in Figure 1. The mean shift based tracker proved to be robust to partial occlusion, clutter, distractors (frame 140 in Figure 1),

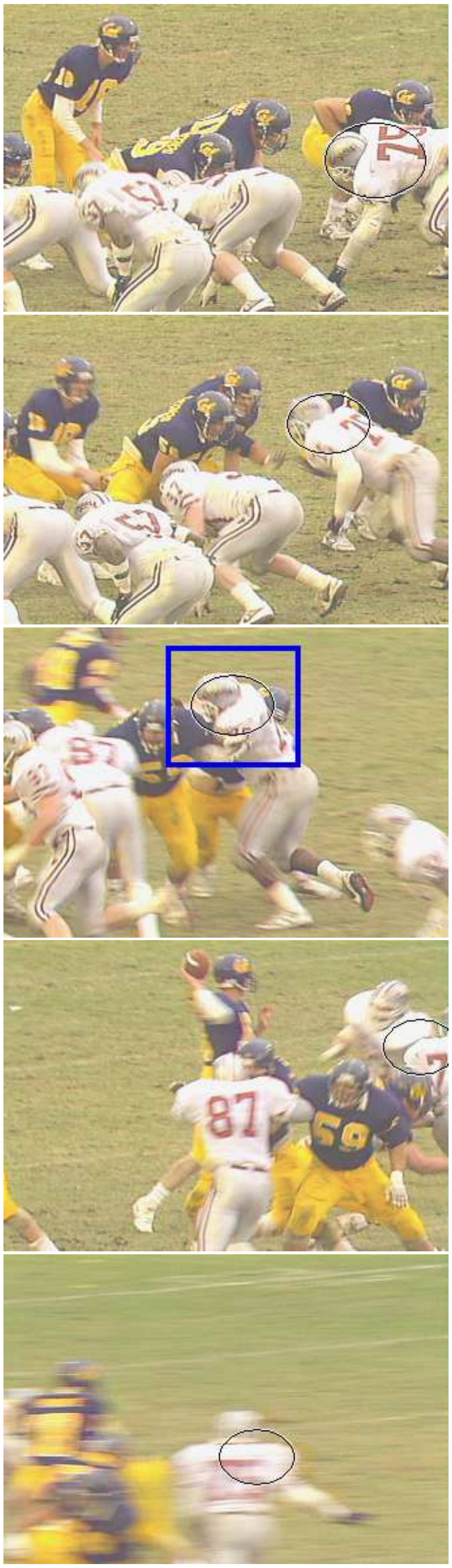


Figure 1: *Football* sequence: Tracking the player no. 75 with initial window of 71×53 pixels. The frames 30, 75, 105, 140, and 150 are shown.

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 and due to the camera motion, did not influence the tracker performance (frame 150 in Figure 1). The same effect, however, can largely perturb contour based trackers.

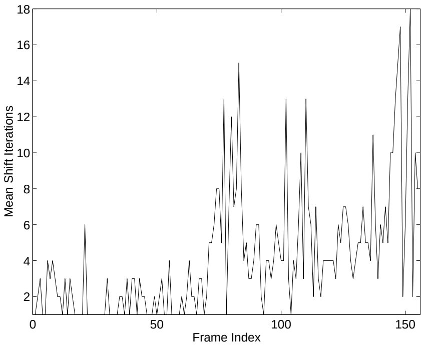


Figure 2: The number of mean shift iterations function of the frame index for the *Football* sequence. The mean number of iterations is 4.19 per frame.

The number of mean shift iterations necessary for each frame (one scale) in the *Football* sequence is shown in Figure 2. One can identify two central peaks, corresponding 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 the fast movement from the left to the right side.

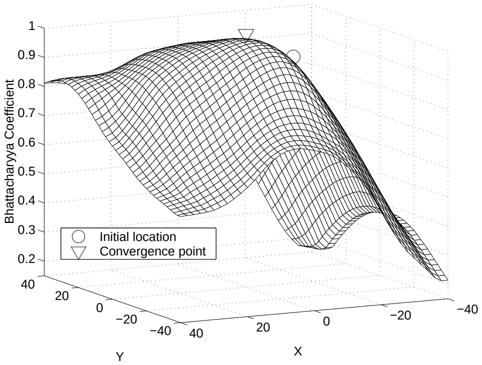


Figure 3: Values of the Bhattacharyya coefficient corresponding to the marked region (81×81 pixels) in frame 105 from Figure 1. The surface is asymmetric, due to the player colors that are similar to the target. Four mean shift iterations were necessary for the algorithm to converge from the initial location (circle).

To demonstrate the efficiency of our approach, Figure 3 presents the surface obtained by computing the Bhattacharyya coefficient for the rectangle marked in Figure 1, frame 105. The target model (the selected elliptical region in frame 30) has been compared with the target candidates obtained by sweeping the elliptical region in frame 105 inside the rectangle. While most of the tracking approaches based on regions [3, 14, 21] 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 since the basin of attraction of the mode covers the entire window, the correct location of the target would have been reached also from farther initial points. An optimized computation of the exhaustive search of the mode [13] has a much larger arithmetic complexity, depending on the chosen search area.

The new method has been applied to track people on subway platforms. The camera being fixed, additional geometric constraints and also background subtraction can be exploited to improve the tracking process. The following sequences, however, have been processed with the algorithm unchanged.

A first example is shown in Figure 4, demonstrating the capability of the tracker to adapt to scale changes. The sequence has 187 frames of 320 × 240 pixels each and the initial normalization constants were $(h_x, h_y) = (23, 37)$.

Figure 5 presents six frames from a 2 minute sequence showing the tracking of a person from the moment she enters the subway platform till she gets on the train (≈ 3600 frames). The tracking performance is remarkable, taking into account the low quality of the processed sequence, due to the compression artifacts. A thorough evaluation of the tracker, however, is subject to our current work.

The minimum value of the distance (18) for each frame is shown in Figure 6. The compression noise determined the distance to increase from 0 (perfect match) to a stationary value of about 0.3. Significant deviations from this value correspond to occlusions generated by other persons or rotations in depth of the target. The large distance increase at the end signals the complete occlusion of the target.

6 Discussion

By exploiting the spatial gradient of the statistical measure (18) the new method achieves real-time tracking performance, while effectively rejecting background clutter and partial occlusions.

Note that the same technique can be employed to derive the measurement vector for optimal prediction schemes such as the (Extended) Kalman filter [1, p.56, 106], or multiple hypothesis tracking approaches [5, 9, 17, 18]. In return, the prediction can determine the priors (defining the presence of the target in a given neighborhood) assumed equal in this paper. This connection is however beyond the scope of this paper. A patent application has been filed covering the tracking algorithm together with the Kalman extension and various applications [29].

We finally observe that the idea of centroid computation is also employed in [22]. The mean shift was used for tracking human faces [4], by projecting the histogram of a face model onto the incoming frame. 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. Gradient based region tracking has been formulated in [2] by minimizing the energy of the deformable region, but no real-time claims were made.

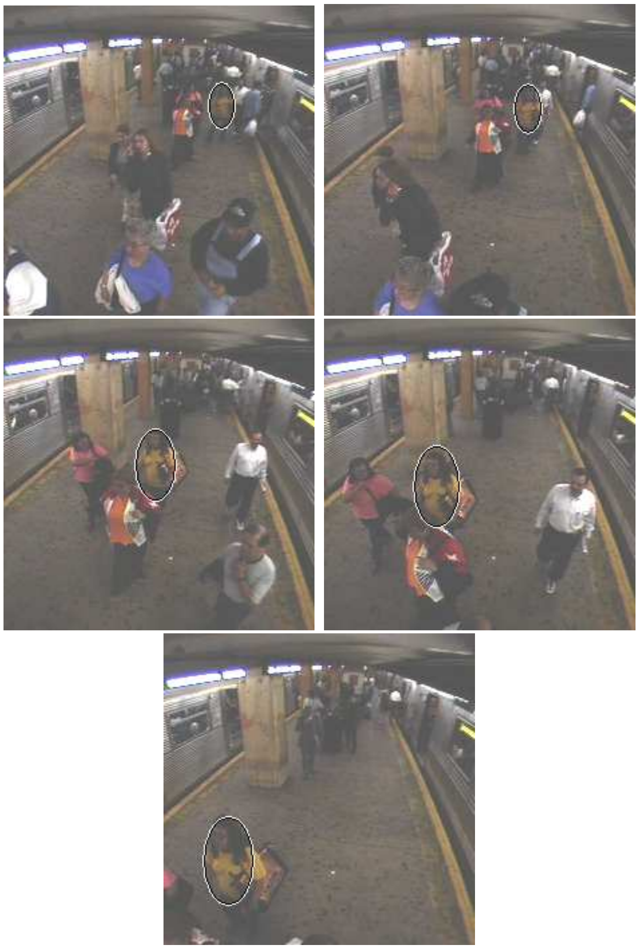


Figure 4: *Subway1* sequence: The frames 500, 529, 600, 633, and 686 are shown (left-right, top-down).

APPENDIX

Proof of Theorem 1

Since n is finite the sequence $\hat{f}_K$ is bounded, therefore, it is sufficient to show that $\hat{f}_K$ is strictly monotonic increasing, i.e., if $\mathbf{y}_j \neq \mathbf{y}_{j+1}$ then $\hat{f}_K(j) < \hat{f}_K(j+1)$, for all $j = 1, 2 \ldots$.

By assuming without loss of generality that $\mathbf{y}_j = \mathbf{0}$ we can write

$$\hat{f}_K(j+1) - \hat{f}_K(j) =$$

$$= \frac{1}{nh^d} \sum_{i=1}^{n} \left[k\left(\left\| \frac{\mathbf{y}_{j+1} - \mathbf{x}_i}{h} \right\|^2 \right) - k\left(\left\| \frac{\mathbf{x}_i}{h} \right\|^2 \right) \right] \quad \text{(A.1)}$$

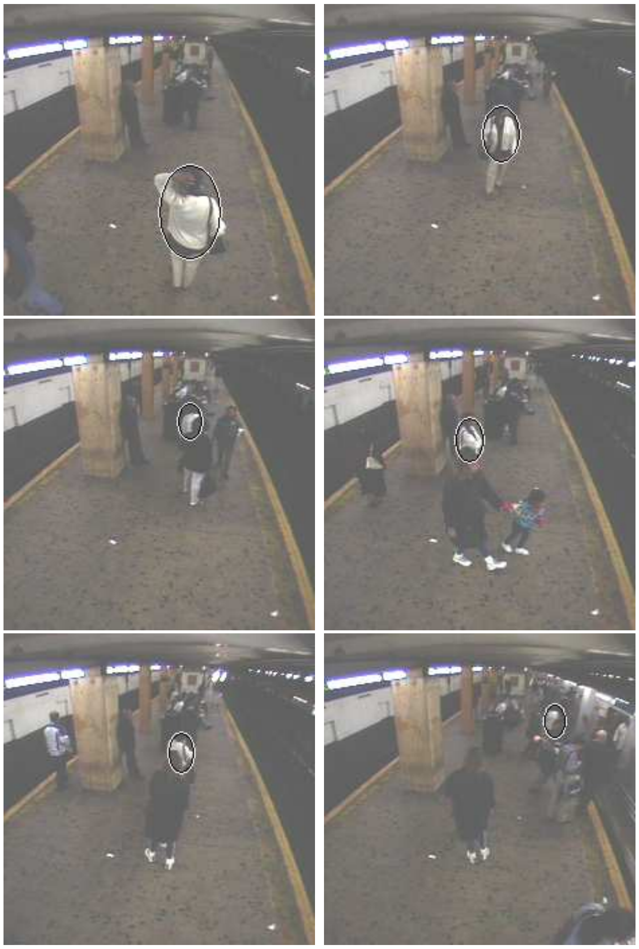


Figure 5: *Subway2* sequence: The frames 3140, 3516, 3697, 5440, 6081, and 6681 are shown (left-right, top-down).

The convexity of the profile k implies that

$$k(x_2) \geq k(x_1) + k'(x_1)(x_2 - x_1) \tag{A.2}$$

for all $x_1, x_2 \in [0, \infty)$, $x_1 \neq x_2$, and since $k' = -g$, the inequality (A.2) becomes

$$k(x_2) - k(x_1) \geq g(x_1)(x_1 - x_2). \tag{A.3}$$

Using now (A.1) and (A.3) we obtain

$$\begin{aligned}
&\hat{f}_K(j+1) - \hat{f}_K(j) \geq \\
&\geq \frac{1}{nh^{d+2}} \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right) \left[\|\mathbf{x}_i\|^2 - \|\mathbf{y}_{j+1} - \mathbf{x}_i\|^2\right] \\
&= \frac{1}{nh^{d+2}} \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right) \left[2\mathbf{y}_{j+1}^\top \mathbf{x}_i - \|\mathbf{y}_{j+1}\|^2\right] = \frac{1}{nh^{d+2}} \\
&\times \left[2\mathbf{y}_{j+1}^\top \sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right) - \|\mathbf{y}_{j+1}\|^2 \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right)\right]
\end{aligned} \tag{A.4}$$

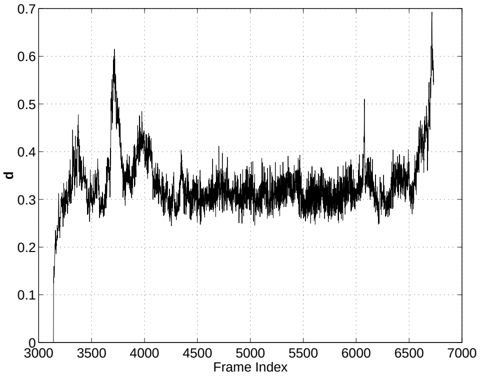


Figure 6: The detected minimum value of distance d function of the frame index for the 2 minute *Subway2* sequence. The peaks in the graph correspond to occlusions or rotations in depth of the target. For example, the peak of value $d \approx 0.6$ corresponds to the partial occlusion in frame 3697, shown in Figure 5. At the end of the sequence, the person being tracked gets on the train, which produces a complete occlusion.

and by employing (14) it results that

$$\hat{f}_K(j+1) - \hat{f}_K(j) \geq \frac{1}{nh^{d+2}} \|\mathbf{y}_{j+1}\|^2 \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right). \tag{A.5}$$

Since k is monotonic decreasing we have $-k'(x) \equiv g(x) \geq 0$ for all $x \in [0, \infty)$. The sum $\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}_i}{h}\right\|^2\right)$ is strictly positive, since it was assumed to be nonzero in the definition of the mean shift vector (10). Thus, as long as $\mathbf{y}_{j+1} \neq \mathbf{y}_j = \mathbf{0}$, the right term of (A.5) is strictly positive, i.e., $\hat{f}_K(j+1) - \hat{f}_K(j) > 0$. Consequently, the sequence $\hat{f}_K$ is convergent.

To prove the convergence of the sequence $\{\mathbf{y}_j\}_{j=1,2\ldots}$ we rewrite (A.5) but without assuming that $\mathbf{y}_j = \mathbf{0}$. After some algebra we have

$$\hat{f}_K(j+1) - \hat{f}_K(j) \geq \frac{1}{nh^{d+2}} \|\mathbf{y}_{j+1} - \mathbf{y}_j\|^2 \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right) \tag{A.6}$$

Since $\hat{f}_K(j+1) - \hat{f}_K(j)$ converges to zero, (A.6) implies that $\|\mathbf{y}_{j+1} - \mathbf{y}_j\|$ also converges to zero, i.e., $\{\mathbf{y}_j\}_{j=1,2\ldots}$ is a Cauchy sequence. This completes the proof, since any Cauchy sequence is convergent in the Euclidean space.

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 (17). According to the Jensen's inequality [8, p.25] we have

$$\rho(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u \hat{q}_u} = \sum_{u=1}^{m} \hat{p}_u \sqrt{\frac{\hat{q}_u}{\hat{p}_u}} \leq \sqrt{\sum_{u=1}^{m} \hat{q}_u} = 1, \tag{A.7}$$

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 $\hat{\mathbf{p}} = \hat{\mathbf{q}}$.

The triangle inequality can be proven as follows. Let us consider the discrete distributions $\hat{\mathbf{p}}$, $\hat{\mathbf{q}}$, and $\hat{\mathbf{r}}$, and define the associated m-dimensional 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, centered at the origin. By taking into account the geometric interpretation of the Bhattacharyya coefficient, the triangle inequality

$$d(\hat{\mathbf{p}},\hat{\mathbf{r}}) + d(\hat{\mathbf{q}},\hat{\mathbf{r}}) \geq d(\hat{\mathbf{p}},\hat{\mathbf{q}}) \tag{A.8}$$

is equivalent to

$$\sqrt{1-cos(\boldsymbol{\xi}_p,\boldsymbol{\xi}_r)}+\sqrt{1-cos(\boldsymbol{\xi}_q,\boldsymbol{\xi}_r)} \geq \sqrt{1-cos(\boldsymbol{\xi}_p,\boldsymbol{\xi}_q)}. \tag{A.9}$$

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.9) is minimized when the vectors $\boldsymbol{\xi}_p$, $\boldsymbol{\xi}_q$, and $\boldsymbol{\xi}_r$ lie in the same plane. Thus, the inequality (A.9) can be reduced to a 2-dimensional problem that can be easily demonstrated by employing the half-angle sinus formula and a few trigonometric manipulations.

Acknowledgement

Peter Meer was supported by the NSF under the grant IRI 99-87695.

References

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

[2] B. Bascle, R. Deriche, "Region Tracking through Image Sequences," *IEEE Int'l Conf. Comp. Vis.*, Cambridge, Massachusetts, 302–307, 1995.

[3] S. Birchfield, "Elliptical Head Tracking using intensity Gradients and Color Histograms," *IEEE Conf. on Comp. Vis. and Pat. Rec.*, Santa Barbara, 232–237, 1998.

[4] G.R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptual User Interface," *IEEE Work. on Applic. Comp. Vis.*, Princeton, 214–219, 1998.

[5] T.J. Cham, J.M. Rehg, "A multiple Hypothesis Approach to Figure Tracking," *IEEE Conf. on Comp. Vis. and Pat. Rec.*, Fort Collins, vol. 2, 239–245, 1999.

[6] D. Comaniciu, P. Meer, "Mean Shift Analysis and Applications," *IEEE Int'l Conf. Comp. Vis.*, Kerkyra, Greece, 1197–1203, 1999.

[7] D. Comaniciu, P. Meer, "Distribution Free Decomposition of Multivariate Data," *Pattern Anal. and Applic.*, 2:22–30, 1999.

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

[9] I.J. Cox, S.L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking," *IEEE Trans. Pattern Analysis Machine Intell.*, 18:138–150, 1996.

[10] Y. Cui, S. Samarasekera, Q. Huang, M Greiffenhagen, "Indoor Monitoring Via the Collaboration Between a Peripheral Senson and a Foveal Sensor," *IEEE Workshop on Visual Surveillance*, Bombay, India, 2–9, 1998.

[11] A. Djouadi, O. Snorrason, F.D. Garber, "The Quality of Training-Sample Estimates of the Bhattacharyya Coefficient," *IEEE Trans. Pattern Analysis Machine Intell.*, 12:92–97, 1990.

[12] A. Eleftheriadis, A. Jacquin, "Automatic Face Location Detection and Tracking for Model-Assisted Coding of Video Teleconference Sequences at Low Bit Rates," *Signal Processing - Image Communication*, 7(3): 231–248, 1995.

[13] F. Ennesser, G. Medioni, "Finding Waldo, or Focus of Attention Using Local Color Information," *IEEE Trans. Pattern Anal. Machine Intell.*, 17(8):805–809, 1995.

[14] P. Fieguth, D. Terzopoulos, "Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates," *IEEE Conf. on Comp. Vis. and Pat. Rec*, Puerto Rico, 21–27, 1997.

[15] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second Ed., Academic Press, Boston, 1990.

[16] S.S. Intille, J.W. Davis, A.F. Bobick, "Real-Time Closed-World Tracking," *IEEE Conf. on Comp. Vis. and Pat. Rec.*, Puerto Rico, 697–703, 1997.

[17] M. Isard, A. Blake, "Condensation - Conditional Density Propagation for Visual Tracking," *Intern. J. Comp. Vis.*, 29(1):5–28, 1998.

[18] M. Isard, A. Blake, "ICondensation: Unifying Low-Level and High-Level Tracking in a Stochastic Framework," *European Conf. Comp. Vision*, Freiburg, Germany, 893–908, 1998.

[19] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Trans. Commun. Tech.*, COM-15:52–60, 1967.

[20] S. Konishi, A.L. Yuille, J. Coughlan, S.C. Zhu, "Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues," *IEEE Conf. on Comp. Vis. and Pat. Rec.*, Fort Collins, 573–579, 1999.

[21] A.J. Lipton, H. Fujiyoshi, R.S. Patil, "Moving Target Classification and Tracking from Real-Time Video," *IEEE Workshop on Applications of Computer Vision*, Princeton, 8–14, 1998.

[22] S.J. McKenna, Y. Raja, S. Gong, "Tracking Colour Objects using Adaptive Mixture Models," *Image and Vision Computing*, 17:223–229, 1999.

[23] N. Paragios, R. Deriche, "Geodesic Active Regions for Motion Estimation and Tracking," *IEEE Int'l Conf. Comp. Vis.*, Kerkyra, Greece, 688–674, 1999.

[24] R. Rosales, S. Sclaroff, "3D trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions," *IEEE Conf. on Comp. Vis. and Pat. Rec.*, Fort Collins, vol. 2, 117–123, 1999.

[25] D.W. Scott, *Multivariate Density Estimation*, New York: Wiley, 1992.

[26] M.J. Swain, D.H. Ballard, "Color Indexing," *Intern. J. Comp. Vis.*, 7(1):11–32, 1991.

[27] P. Viola, W.M. Wells III, "Alignment by Maximization of Mutual Information," *IEEE Int'l Conf. Comp. Vis.*, Cambridge, Massachusetts, 16–23, 1995.

[28] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis Machine Intell.*, 19:780–785, 1997.

[29] "Real-Time Tracking of Non-Rigid Objects using Mean Shift," US patent pending.

ARTICLE IN PRESS

ELSEVIER

Image and Vision Computing xx (2002) 1–12

image AND vision COMPUTING

www.elsevier.com/locate/imavis

An adaptive color-based particle filter

Katja Nummiaro[a,*], Esther Koller-Meier[b], Luc Van Gool[a,b]

[a] *Katholieke Universiteit Leuven, ESAT/PSI-VISICS, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium*
[b] *Swiss Federal Institute of Technology (ETH), D-ITET/BIWI, Gloriastrasse 35, 8092 Zurich, Switzerland*

Abstract

Robust real-time tracking of non-rigid objects is a challenging task. Particle filtering has proven very successful for non-linear and non-Gaussian estimation problems. The article presents the integration of color distributions into particle filtering, which has typically been used in combination with edge-based image features. Color distributions are applied, as they are robust to partial occlusion, are rotation and scale invariant and computationally efficient. As the color of an object can vary over time dependent on the illumination, the visual angle and the camera parameters, the target model is adapted during temporally stable image observations. An initialization based on an appearance condition is introduced since tracked objects may disappear and reappear. Comparisons with the mean shift tracker and a combination between the mean shift tracker and Kalman filtering show the advantages and limitations of the new approach.

Keywords: Particle filtering; Condensation algorithm; Color distribution; Bhattacharyya coefficient; Mean shift tracker

1. Introduction

Object tracking is required by many vision applications such as human-computer interfaces [2], video communication/compression [22] or surveillance [3,9,27]. In this context, particle filters provide a robust tracking framework as they are neither limited to linear systems nor require the noise to be Gaussian.

The idea of a particle filter–to apply a recursive Bayesian filter based on sample sets–was independently proposed by several research groups [8,12,14,18]. Our work has evolved from the Condensation algorithm [12,14] which was developed in the computer vision community and was typically used with edge-based image features [11,12,14,20] . At the same time this filtering method was studied both in statistics and signal processing known in that context as Bayesian bootstrap filter [8] or Monte Carlo Filter [18].

We propose to use such a particle filter with color-based image features. Color histograms in particular have many advantages for tracking non-rigid objects as they are robust to partial occlusion, are rotation and scale invariant and are calculated efficiently. A target is tracked with a particle filter by comparing its histogram with the histograms of the sample positions using the Bhattacharyya distance. Fig. 1 shows the application of the color-based particle filter for tracking the face of a soccer player.

The novelty of the proposed approach mainly lies in the original mixture of efficient components that together yield a reliable and fast tracking performance for non-rigid objects.

In general, tracking methods can be divided into two main classes specified as *bottom-up* or *top-down* approaches. In a *bottom-up* approach the image is segmented into objects which are then used for the tracking. For example blob detection [19] can be used for the object extraction. In contrast, a *top-down* approach generates object hypotheses and tries to verify them using the image. Typically, model-based [12,14] and template matching approaches [5] belong to this class. The proposed color-based particle filter follows the *top-down* approaches, in the sense that the image content is only evaluated at the sample positions.

The related mean shift tracker by Comaniciu et al. [5] also uses color distributions. By employing multiple hypotheses and a model of the system dynamics our proposed method can track objects more reliably in cases of clutter and occlusions. Jepson et al., McKenna et al. and Raja et al. [16,21,26] have already discussed adaptive models, but these approaches employ Gaussian mixture models while we use color histograms together with multiple hypotheses. Isard et al. [15] have already employed color information in particle filtering by using

[*] Corresponding author. Tel.: +32-16-321061; fax: +32-16-321723.
E-mail address: knummiar@esat.kuleuven.ac.be (K. Nummiaro).

0262-8856/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0262-8856(02)00129-4

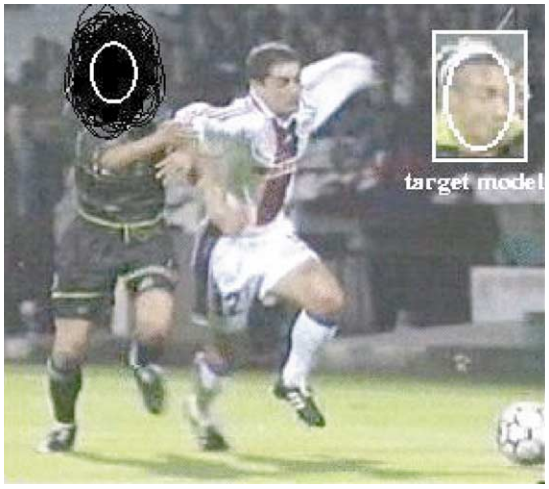


Fig. 1. A color-based target model and the different hypotheses (black ellipses) calculated with the particle filter. The white ellipse on the left represents the expected object location.

Gaussian mixtures. In comparison, our target model has the advantage of matching only objects that have a similar histogram, whereas for Gaussian mixtures objects that contain one of the colors of the mixture will already match. Recently, Pérez et al. [25] introduced an approach that also uses color histograms and a particle filtering framework for multiple object tracking. The two independently proposed methods differ in the initialization of the tracker, the model update, the region shape and the observation of the tracking performance.

The outline of this article is as follows. In Section 2 we briefly describe particle filtering and in Section 3 we indicate how color distributions are used as object models. The integration of the color information into the particle filter is explained in Section 4 and Section 5 describes the model update. As tracked objects may disappear and reappear an initialization based on an appearance condition is introduced in Section 6. Section 7 compares the mean shift [5] and the Kalman/mean shift tracker [6] with our proposed tracking framework. In Section 8 we present some experimental results and finally, in Section 9, we summarize our conclusions.

2. Particle filtering

Particle filtering [12,14] was originally developed to track objects in clutter. The state of a tracked object is described by the vector X_t while the vector Z_t denotes all the observations $\{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ up to time t. Particle filters are often used when the posterior density $p(X_t|Z_t)$ and the observation density $p(\mathbf{z}_t|X_t)$ are non-Gaussian.

The key idea of particle filtering is to approximate the probability distribution by a weighted sample set $S = \{(\mathbf{s}^{(n)}, \pi^{(n)})|n = 1\ldots N\}$. Each sample $\mathbf{s}$ represents one hypothetical state of the object, with a corresponding discrete sampling probability π, where $\sum_{n=1}^{N} \pi^{(n)} = 1$.

The evolution of the sample set is described by propagating each sample according to a system model. Each element of the set is then weighted in terms of the observations and N samples are drawn with replacement, by choosing a particular sample with probability $\pi^{(n)} = p(\mathbf{z}_t|X_t = \mathbf{s}_t^{(n)})$. The mean state of an object is estimated at each time step by

$$E[S] = \sum_{n=1}^{N} \pi^{(n)} \mathbf{s}^{(n)}. \tag{1}$$

Particle filtering provides a robust tracking framework, as it models uncertainty. It can keep its options open and consider multiple state hypotheses simultaneously. Since less likely object states have a chance to temporarily remain in the tracking process, particle filters can deal with short-lived occlusions.

3. Color distribution model

We want to apply a particle filter in a color-based context. Color distributions are used as target models as they achieve robustness against non-rigidity, rotation and partial occlusion. Suppose that the distributions are discretized into m-bins. The histograms are produced with the function $h(\mathbf{x}_i)$, that assigns the color at location $\mathbf{x}_i$ to the corresponding bin. In our experiments, the histograms are typically calculated in the RGB space using $8 \times 8 \times 8$ bins. To make the algorithm less sensitive to lighting conditions, the HSV color space could be used instead with less sensitivity to V (e.g. $8 \times 8 \times 4$ bins).

We determine the color distribution inside an upright elliptic region with half axes H_x and H_y. To increase the reliability of the color distribution when boundary pixels belong to the background or get occluded, smaller weights are assigned to the pixels that are further away from the region center by employing a weighting function

$$k(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where r is the distance from the region center. Thus, we increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded. It is also possible to use a different weighting function, for example the Epanechnikov kernel [5]. The color distribution $p_{\mathbf{y}} = \{p_{\mathbf{y}}^{(u)}\}_{u=1\ldots m}$ at location $\mathbf{y}$ is calculated as

$$p_{\mathbf{y}}^{(u)} = f \sum_{i=1}^{I} k\left(\frac{\|\mathbf{y} - \mathbf{x}_i\|}{a}\right) \delta[h(\mathbf{x}_i) - u] \tag{3}$$

where I is the number of pixels in the region, δ is the Kronecker delta function, the parameter $a = \sqrt{H_x^2 + H_y^2}$ is used to adapt the size of the region, and the normalization factor

$$f = \frac{1}{\sum_{i=1}^{I} k\left(\frac{\|\mathbf{y} - \mathbf{x}_i\|}{a}\right)} \tag{4}$$

ensures that $\sum_{u=1}^{m} p_{\mathbf{y}}^{(u)} = 1$.

In a tracking approach, the estimated state is updated at each time step by incorporating the new observations. Therefore, we need a similarity measure, which is based on color distributions. A popular measure between two distributions $p(u)$ and $q(u)$ is the Bhattacharyya coefficient [1,17]

$$\rho[p,q] = \int \sqrt{p(u)q(u)}du. \tag{5}$$

Considering discrete densities such as our color histograms $p = \{p^{(u)}\}_{u=1\ldots m}$ and $q = \{q^{(u)}\}_{u=1\ldots m}$ the coefficient is defined as

$$\rho[p,q] = \sum_{u=1}^{m} \sqrt{p^{(u)}q^{(u)}}. \tag{6}$$

The larger ρ is, the more similar the distributions are. For two identical normalized histograms we obtain $\rho = 1$, indicating a perfect match. As distance between two distributions we define the measure

$$d = \sqrt{1 - \rho[p,q]} \tag{7}$$

which is called the Bhattacharyya distance.

4. Color-based particle filtering

The proposed tracker employs the Bhattacharyya distance to update the a priori distribution calculated by the particle filter. Each sample of the distribution represents an ellipse and is given as

$$\mathbf{s} = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{a}\} \tag{8}$$

where x, y specify the location of the ellipse, $\dot{x}, \dot{y}$ the motion, H_x, H_y the length of the half axes and $\dot{a}$ the corresponding scale change. As we consider a whole sample set the tracker handles multiple hypotheses simultaneously.

The sample set is propagated through the application of a dynamic model

$$\mathbf{s}_t = A\mathbf{s}_{t-1} + \mathbf{w}_{t-1} \tag{9}$$

where A defines the deterministic component of the model and $\mathbf{w}_{t-1}$ is a multivariate Gaussian random variable. In our application we currently use a first order model for A describing a region moving with constant velocity $\dot{x}, \dot{y}$ and scale change $\dot{a}$. Expanding this model to second order is straightforward.

To weight the sample set, the Bhattacharyya coefficient has to be computed between the target histogram and the histogram of the hypotheses. Each hypothetical region is specified by its state vector $\mathbf{s}^{(n)}$. Both the target histogram q and the candidate histogram $p_{\mathbf{s}^{(n)}}$ are calculated from Eq. (3) where the target is centered at the origin of the elliptic region.

As we want to favor samples whose color distributions are similar to the target model, small Bhattacharyya distances correspond to large weights:

$$\pi^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(1-\rho[p_{\mathbf{s}^{(n)}},q])}{2\sigma^2}} \tag{10}$$

that are specified by a Gaussian with variance σ. During filtering, samples with a high weight may be chosen several times, leading to identical copies, while others with relatively low weights may not be chosen at all. The programming details for one iteration step are given in Fig. 2. The proposed color-based particle filter was introduced in [23,24].

To illustrate the distribution of the sample set, Fig. 3 shows the Bhattacharyya coefficient for a rectangular region of the soccer player shown in Fig. 1. The samples are located around the maximum of the Bhattacharyya coefficient, which represents the best match to the target model. As can be seen, the calculated mean state of the sample distribution corresponds well to the maximum and consequently the localization of the face is accurate.

5. Target model update

Illumination conditions, the visual angle, as well as the camera parameters can influence the quality of the color-based particle filter. To overcome the resulting appearance

```
Given the sample set $S_{t-1}$ and the target model
$q = f \sum_{i=1}^{I} k\left(\frac{\|\mathbf{x}_i\|}{a}\right) \delta[h(\mathbf{x}_i) - u]$,
perform the following steps:
  (1) Select $N$ samples from the set $S_{t-1}$ with probability $\pi_{t-1}^{(n)}$:
      (a) calculate the normalized cumulative probabilities $c'_{t-1}$
            $c_{t-1}^{(0)} = 0$
            $c_{t-1}^{(n)} = c_{t-1}^{(n-1)} + \pi_{t-1}^{(n)}$
            $c_{t-1}^{\prime(n)} = \frac{c_{t-1}^{(n)}}{c_{t-1}^{(N)}}$
      (b) generate a uniformly distributed random number $r \in [0,1]$
      (c) find, by binary search, the smallest $j$ for which $c_{t-1}^{\prime(j)} \geq r$
      (d) set $\mathbf{s}_{t-1}^{\prime(n)} = \mathbf{s}_{t-1}^{(j)}$
  (2) Propagate each sample from the set $S'_{t-1}$ by a linear stochastic
      differential equation:
      $\mathbf{s}_t^{(n)} = A\,\mathbf{s}_{t-1}^{\prime(n)} + \mathbf{w}_{t-1}^{(n)}$
      where $\mathbf{w}_{t-1}^{(n)}$ is a multivariate Gaussian random variable
  (3) Observe the color distributions:
      (a) calculate the color distribution
            $p_{\mathbf{s}_t^{(n)}}^{(u)} = f \sum_{i=1}^{I} k\left(\frac{\left\|\mathbf{s}_t^{(n)} - \mathbf{x}_i\right\|}{a}\right) \delta[h(\mathbf{x}_i) - u]$
          for each sample of the set $S_t$
      (b) calculate the Bhattacharyya coefficient for each sample
          of the set $S_t$
          $\rho[p_{\mathbf{s}_t^{(n)}}, q] = \sum_{u=1}^{m} \sqrt{p_{\mathbf{s}_t^{(n)}}^{(u)} q^{(u)}}$
      (c) weight each sample of the set $S_t$
          $\pi_t^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(1-\rho[p_{\mathbf{s}_t^{(n)}},q])}{2\sigma^2}}$
  (4) Estimate the mean state of the set $S_t$
      $E[S_t] = \sum_{n=1}^{N} \pi_t^{(n)} \mathbf{s}_t^{(n)}$
```

Fig. 2. An iteration step of the color-based particle filter.

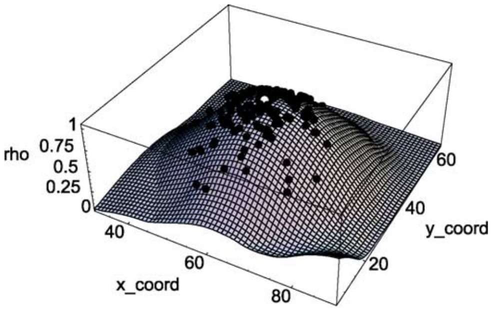


Fig. 3. Surface plot of the Bhattacharyya coefficient of a small area around the face of the soccer player shown in Fig. 1. The black points illustrate the centers of the ellipses of the sample set while the white point represents the mean location. It is positioned close to the maximum of the plot.

changes we update the target model during slowly changing image observations. By discarding image outliers–where the object is occluded or too noisy–it can be ensured that the model is not updated when the tracker has lost the object. So, we use the update condition

$$\pi_{E[S]} > \pi_T \tag{11}$$

where $\pi_{E[S]}$ is the observation probability of the mean state $E[S]$ and π_T is a threshold.

The update of the target model is implemented by the equation

$$q_t^{(u)} = (1 - \alpha)q_{t-1}^{(u)} + \alpha p_{E[S_t]}^{(u)} \tag{12}$$

for each bin u where α weights the contribution of the mean state histogram $p_{E[S_t]}$. Thus, we evoke a forgetting process in the sense that the contribution of a specific frame decreases exponentially the further it lies in the past. A similar approach is often used for model updates in figure-background segmentation algorithms [7,10].

To summarize, one single target model is used, respectively adapted, for the whole sample set of the particle filter. We have also considered to use different target models for each sample but the computational cost increases while the results are not significantly better. Furthermore, some samples could adapt themselves to a wrong target.

6. Initialization

For the initialization of the particle filter, we have to find the initial starting values x, y, H_x and H_y. There are three possibilities depending on the prior knowledge of the target object: manual initialization, automatic initialization using a known histogram as target model or an object detection algorithm that finds interesting targets. Whatever the choice, the object must be fully visible, so that a good color distribution can be calculated.

If the target histogram $q = \{q^{(u)}\}_{u=1...m}$ is known, we can place samples strategically at positions where the target is expected to appear (Fig. 4). The tracker should detect the object when it enters the field of view of the camera. In this case, the Bhattacharyya coefficient in the vicinity of the object position should be significantly higher than the average coefficient of the background. Therefore, we first calculate the mean value μ and the standard deviation σ of the Bhattacharyya coefficient for elliptic regions over all the positions of the background:

$$\mu = \frac{1}{I}\sum_{i=1}^{I} \rho[p_{\mathbf{x}_i}, q] \tag{13}$$

$$\sigma^2 = \frac{1}{I}\sum_{i=1}^{I} (\rho[p_{\mathbf{x}_i}, q] - \mu)^2 \tag{14}$$

and then define an appearance condition as

$$\rho[p_{\mathbf{s}_t^{(n)}}, q] > \mu + 2\sigma. \tag{15}$$

This indicates a 95% confidence that a sample does not belong to the background. If more than $b{\cdot}N$ of the samples fulfill the appearance condition during initialization, we consider the object to be found and start tracking. The parameter b is called the ‘kick-off fraction’.

Likewise, the same condition is used to determine if an object is lost during the tracking. If the number of positive appearances is smaller than $b{\cdot}N$ for a couple of frames, the tracker returns into the ‘initialization’ mode. In our experiments a value of $b = 0.1$ has been proven sufficient.

In several of the experiments, the goal was to track faces, and we used an automatic object detection algorithm based on Support Vector Machines [4] for the initialization.

7. Comparisons

The mean shift algorithm has been introduced recently for tracking and segmentation applications [5,28]. It is a simple and fast adaptive tracking procedure that finds the maximum of the Bhattacharyya coefficient given a target model and a starting region. Based on the mean shift vector, which is an estimation of the gradient of the Bhattacharyya

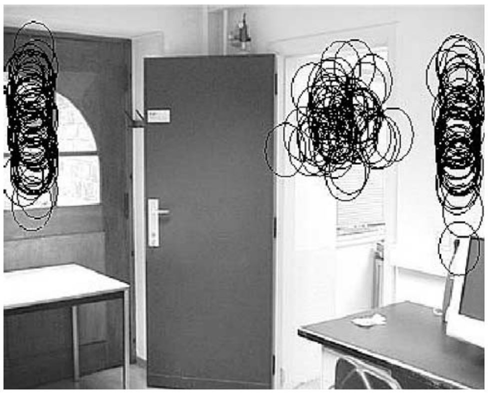


Fig. 4. Example from the *surveillance* experiment (see Section 8). The samples are initially placed at positions where the known human head is most likely to appear, like doors and image borders.

function, the new object location is calculated. This step is repeated until the location no longer changes significantly. A target scaling is taken into account by calculating the Bhattacharyya coefficient for three different sizes (same scale, ±5% change) and choosing the size which gives the highest similarity to the target model.

To reduce the number of iterations for the best object location the basic mean shift tracker was enhanced by a state prediction using Kalman filtering [6]. If a Kalman filter is used to estimate the new location, the search regions of subsequent frames no longer need to overlap and the tracker is more likely to converge to the correct maxima in case of rapid movements.

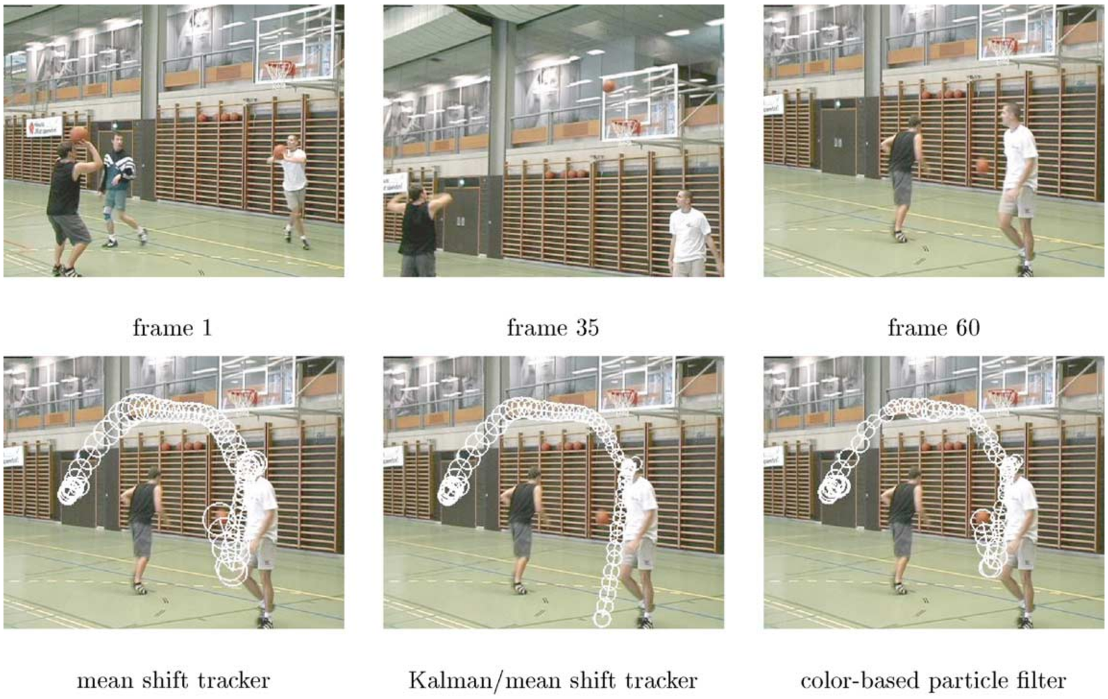
To illustrate the differences between the mean shift, the Kalman/mean shift tracker and our proposal we discuss a *basketball* and a *snowboarder* sequences. The experiments have been processed with a Pentium3 800 MHz PC under Linux, using the RGB color space with 8 × 8 × 8 bins.

In the *basketball* sequence (Fig. 5) the ball is thrown into the basket, afterwards bouncing from the floor again. The results of the three trackers (see bottom row) are illustrated by the paths of the elliptic regions. The image size is 360 × 288 pixels and the initial elliptic search region contains 20 × 20 pixels. As can be seen from the left image, the mean shift tracker can trace the ball during the whole sequence but does not always detect the correct scale. The mean shift iteration itself has no integrated scale adaptation. As mentioned before, scale is handled by calculating the Bhattacharyya distance with different fixed scales in order to detect possible size changes during the sequence. In this example the mean shift tracker chooses a large enough region so that the search regions still overlap despite of the fast movement of the basketball. Consequently, the search region increases although the target size stays constant. If no scaling is employed or the maximum scale change is too small, the mean shift tracker loses the ball.

In contrast, the Kalman/mean shift tracker (see middle image) can handle the scaling better due to the prediction of the search region. However, the state estimation proves false during the bounce and consequently the tracker loses the ball.

Finally, for the color-based particle filter (see right image) we processed $N = 75$ samples. In comparison to the mean shift tracker the scaling results look better but are less smooth as for the Kalman/mean shift tracker. An improvement can be achieved by increasing the number of samples but this affects the computational performance. The color-based particle filter predicts the search region similarly to the Kalman/mean shift tracker but it can still track the ball after bouncing from the floor due to its multiple hypotheses. To increase the flexibility of the color-based particle filter, it can be further enhanced by switching between multiple motion models [13].

In the *snowboarder* sequence (Fig. 6) the goal is to follow the boarder during his jump. The image size is 352 × 240 pixels and the dimensions of the initial elliptic search region is 14 × 22 pixels. One of the biggest problems in motion-based tracking is to follow an object through clutter. Such a situation is shown in frame 50 where for example the mean shift tracker (see second row) converges to a local maximum, which corresponds to a region in


frame 1

frame 35

frame 60

mean shift tracker

Kalman/mean shift tracker

color-based particle filter

Fig. 5. Top row: The sequence of the ball thrown into the basket; Bottom row: The results of the mean shift tracker, the Kalman/mean shift tracker and the color-based particle filter.

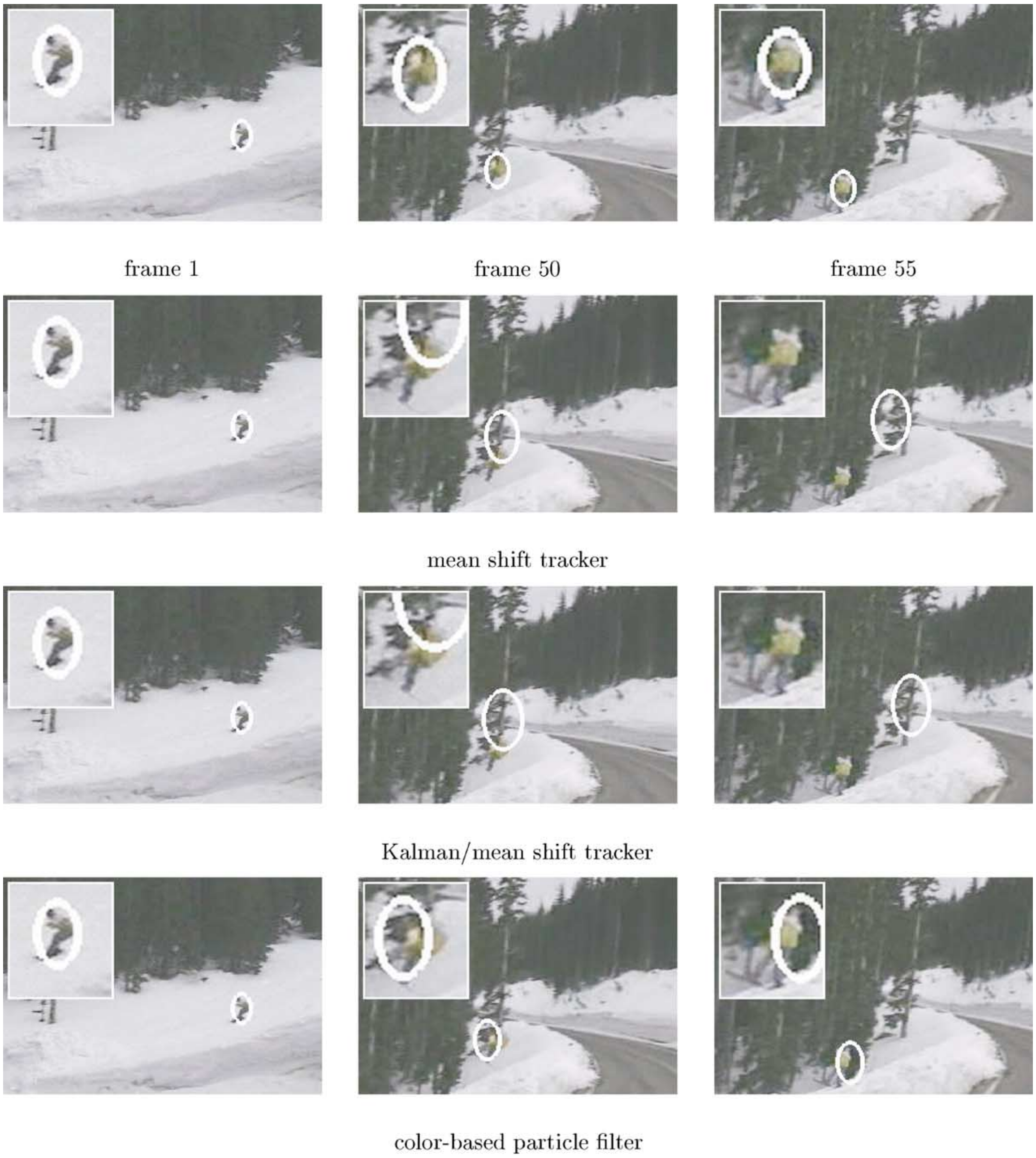


Fig. 6. The top row shows the sequence of a snowboarder jumping. The second row displays the results of the mean shift tracker, the third row shows the output of the combination of the mean shift tracker with the Kalman filter and in the last row the results of the color-based particle filter are shown.

the background that has a similar color distribution as the target model. In this situation the tracker has no chance to recover and must be re-initialized. If a Kalman filter is used to estimate the new location for the mean shift iterations (see third row), the situation looks similar as the state prediction does not correspond well to the observation. The tracker still has problems to follow the object through clutter as a single hypothesis is used for the tracking. In contrast, the color-based particle filter (see last row) tracks multiple hypotheses and is therefore more reliable.

In summary, first, the mean shift iteration itself has no scale adaptation while in the color-based particle filtering the scale is directly estimated and propagated using the system model. Consequently, the scale changes freely, adapts better to the actual size of the object and is more accurate.

Secondly, a state prediction can improve the tracking results of the mean shift approach in case of rapid movements, but the system dynamics must represent the object's movement well enough. However, a single hypothesis still limits the flexibility of the tracker in case of clutter.

Thirdly, the mean shift and the Kalman/mean shift tracker have the advantage that a more precise localization

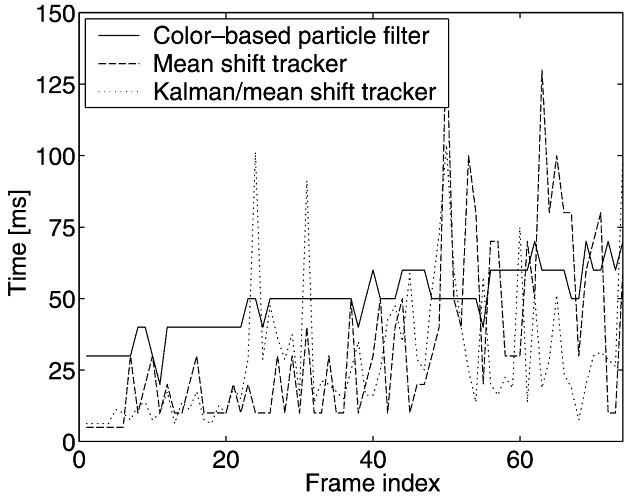


Fig. 7. Running times between two successive frames of the *snowboarder* sequence.

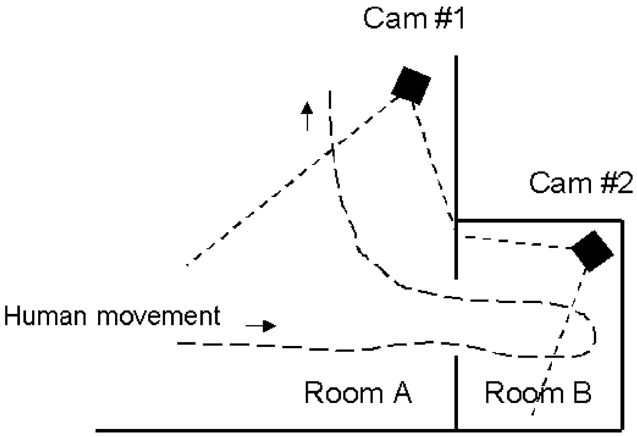


Fig. 8. Camera setup and the person's path in the *surveillance* example.

is calculated, which corresponds to a maximum of the similarity measure. In particle filtering, the object location has to be estimated by calculating the mean value of the sample distribution. Accordingly, the accuracy of the tracker is dependent on the size of the sample set. By increasing the number of samples the discretization error can be decreased.

The running time to process one frame depends mainly on the region size for all approaches as many color distributions have to be calculated. When using the mean shift or the Kalman/mean shift tracker the number of these calculations depend on the number of iterations, and with particle filtering on the number of samples. If the system model represents the object movement well enough, the Kalman state prediction can reduce the number of mean shift iterations. The computing times of all trackers are shown in Fig. 7 for the *snowboarder* sequence. On the average, the mean shift tracker and the Kalman/mean shift tracker are faster but they need more computation time in frames where they lose the object. However, all trackers have real time capabilities.

8. More results

We consider a mock *surveillance* sequence of 450 frames to demonstrate the efficiency of the color-based particle filter. The system uses two fixed cameras to track a person who is moving inside two connected rooms. The ground plan is shown in Fig. 8. The cameras are kept static without any zoom, pan or tilt and their relative exterior orientation is known. Camera 1 in room A is pointed to a door, which leads to a room B that is observed by camera 2. Currently, the trackers in both cameras are working independently, i.e.

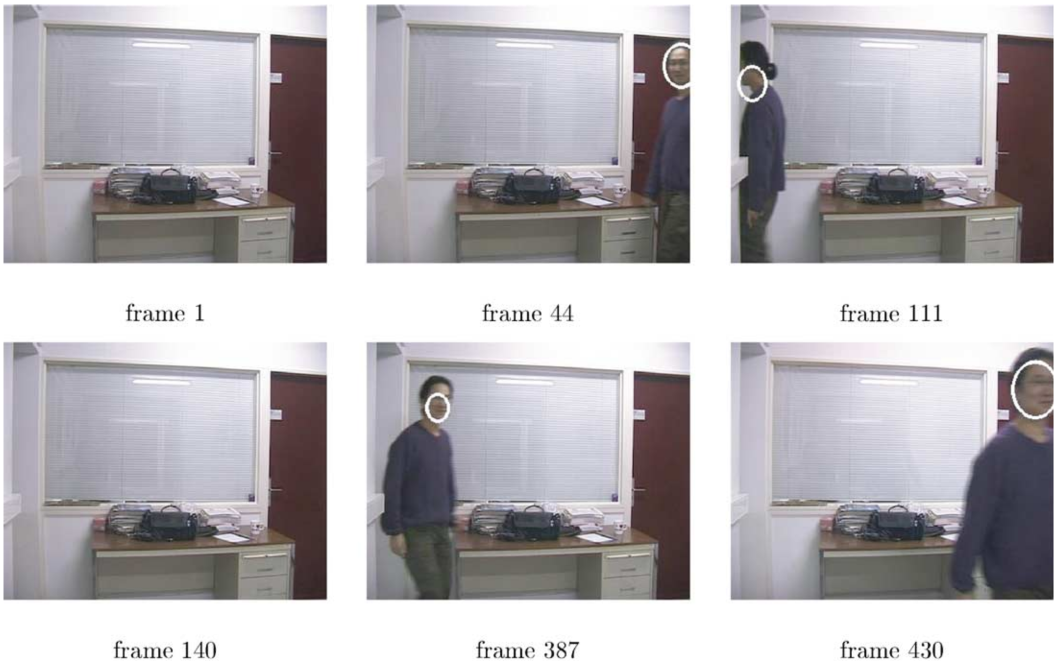


Fig. 9. Frame selection from the *surveillance* sequence of camera 1.

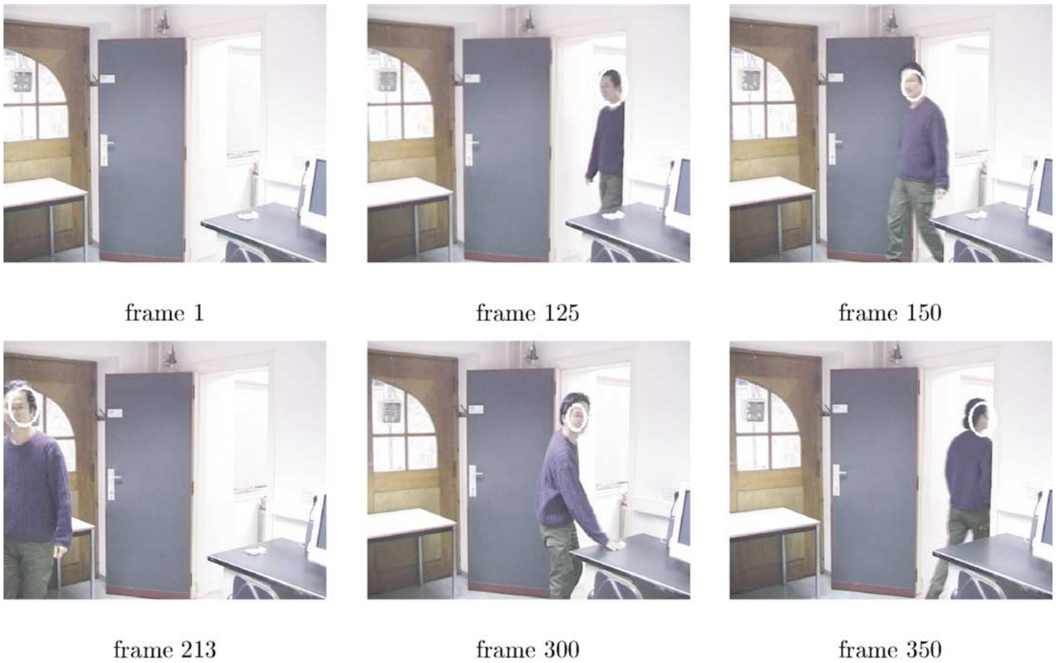


Fig. 10. Frame selection from the *surveillance* sequence of camera 2.

each of them uses a separate particle filter and do not exchange information.

In this experiment we used the initialization method based on a known histogram. Both trackers are put into the 'initialization' mode and start tracking as soon as a person enters their field of view. When the person later leaves the room, the corresponding tracker will return to the 'initialization' mode. In Figs. 9 and 10 the results of camera 1 and camera 2 are shown. The trackers handle the initialization successfully, even when the person is appearing from different sides. Also scale changes and out-of-plane rotations are managed properly. In particular, the rotations are very large as the face is seen from the front as well as from the side.

The method could be further improved by letting the trackers exchange information. For example when a person leaves room B, the exact position, velocity and region size could be handed over to the tracker in room A which can then initialize a sample distribution using this knowledge.

Switching between the 'initialization' and 'tracking' modes is done by applying the appearance condition of Eq. (15). Fig. 11 shows the number of positive appearances for both cameras. In this experiment we used $N = 100$ samples and the 'kick-off' fraction $b = 0.1$.

To demonstrate the robustness of our color-based particle filter against occlusion and rapid movements, we show results for a *soccer* sequence, where the tracker follows a single player over 438 frames. The results are displayed in Fig. 12. In this sequence the camera is moving. The player is completely occluded by the referee in frame 156, but despite of other good object candidates in the neighborhood, the particle filter performs perfectly. Small gaps during tracking can occur when the occlusion continues for a longer period. In these cases, the mean state is not located very accurately for a short time, but due to multiple hypotheses the tracker can recover the player. Fig. 13 shows the evolution of the size and position of the object region during the *soccer* sequence. As can be seen, the scale changes quite smoothly.

In Fig. 14 we consider a *moving stairs* sequence in a train station. A static surveillance camera is installed to track the faces of passing passengers. In this experiment the robustness of color-based particle filtering against occlusion and large scale changes is demonstrated. During the whole sequence the tracker has to cope with a large scale change as the person is approaching the surveillance camera. In frame 19, the object is temporarily lost as it is completely occluded, but can be recovered using the appearance condition given in Eq. (15). The new initial location is poor at the beginning but improves quickly after a few frames due to the use of multiple hypotheses.

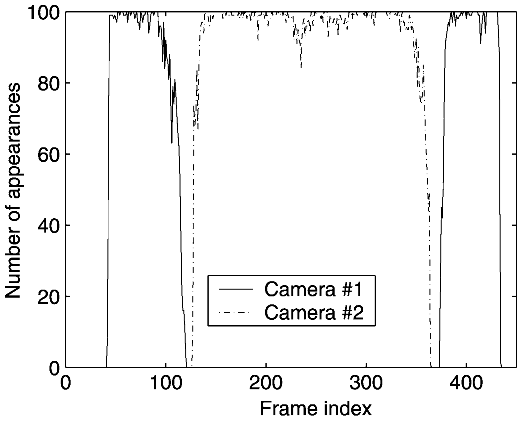
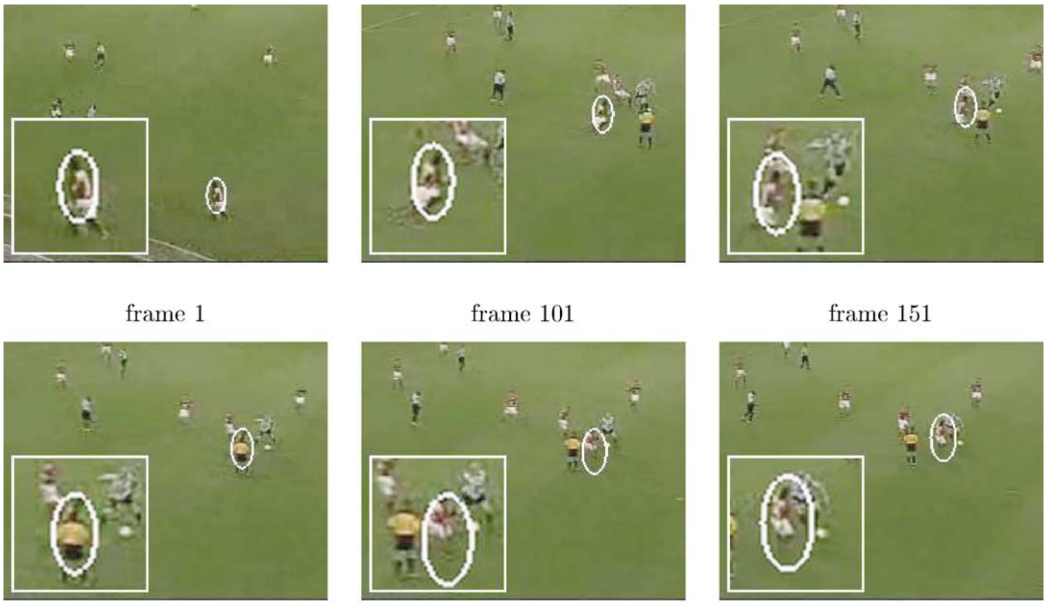


Fig. 11. The number of samples in each frame that fulfill the appearance condition for the *surveillance* sequence.



frame 1 frame 101 frame 151

frame 156 frame 161 frame 166

Fig. 12. This *soccer* sequence shows the successful tracking of a player in cases of occlusion and rapid movement.

To corroborate the importance of the model update we consider the *traffic* sequence of 234 frames recorded by a highway monitoring system. There is an evident scale change during this sequence as the camera was placed towards the traffic flow. Furthermore, different viewing angles of the car and partial occlusions make the experiment more difficult. In the top row of Fig. 15 no model update is performed and the resulting region gets stuck on the left front side of the car. In contrast, the bottom row shows the effectiveness of the model update as now also the scale is getting adapted correctly.

Fig. 16 shows the *face* sequence of 600 frames, taken under strong lightning changes by the sun. At the beginning of the sequence the face is in the shadow of the trees and at the end it is directly in the sun. The tracked face is affected by changing illumination conditions and facial expressions as well as a full turn of the person and large scale changes. In frame 400, the tracked position is not very exact as the model does not match the back of the head very well. Nevertheless, the person can still be tracked and the position improves rapidly once the person has turned around.

The target model of our tracker is only updated according to Eq. (11) as outliers must be discarded, i.e. the update is only made when the object is tracked stably. A related update condition is given by the maximization of the log-likelihood [16] over the last T frames: $L = \sum_{t=1}^{T} log \pi_{E[S]}^{(t)}$. In Fig. 17 both update possibilities are plotted for the *face* sequence. The two update approaches behave similarly in the sense that a model update is only performed under slowly varying image conditions. As the history of samples through the log-likelihood does not significantly improve the results, we use our simpler and therefore more efficient method.

9. Conclusions

The proposed tracking method adds an adaptive appearance model based on color distributions to particle filtering. The color-based tracker can efficiently and successfully handle non-rigid and fast moving objects under different

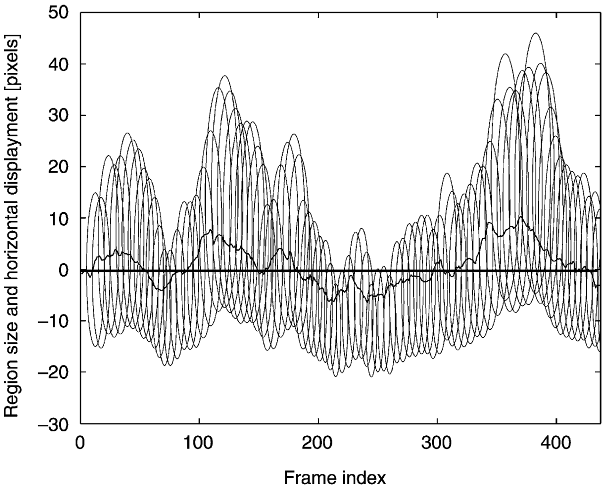
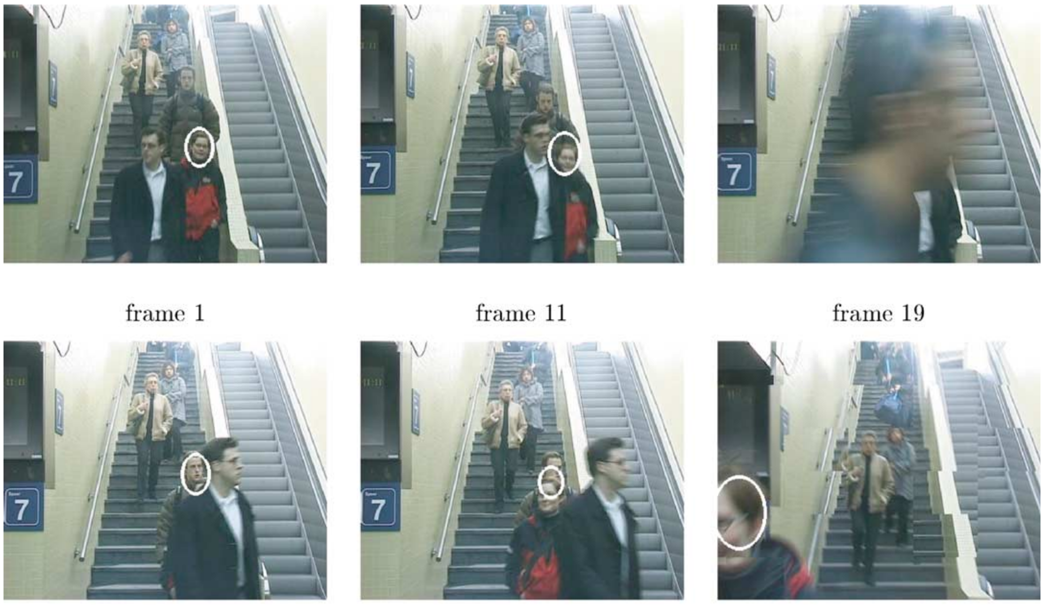


Fig. 13. The size and horizontal position of the elliptic object region during the *soccer* sequence is shown for every fifth frame. The scaling changes relatively smoothly. The line indicates the horizontal displacement which is relatively small as the player stays more or less in the center of the field of view, as a movable camera has been used.



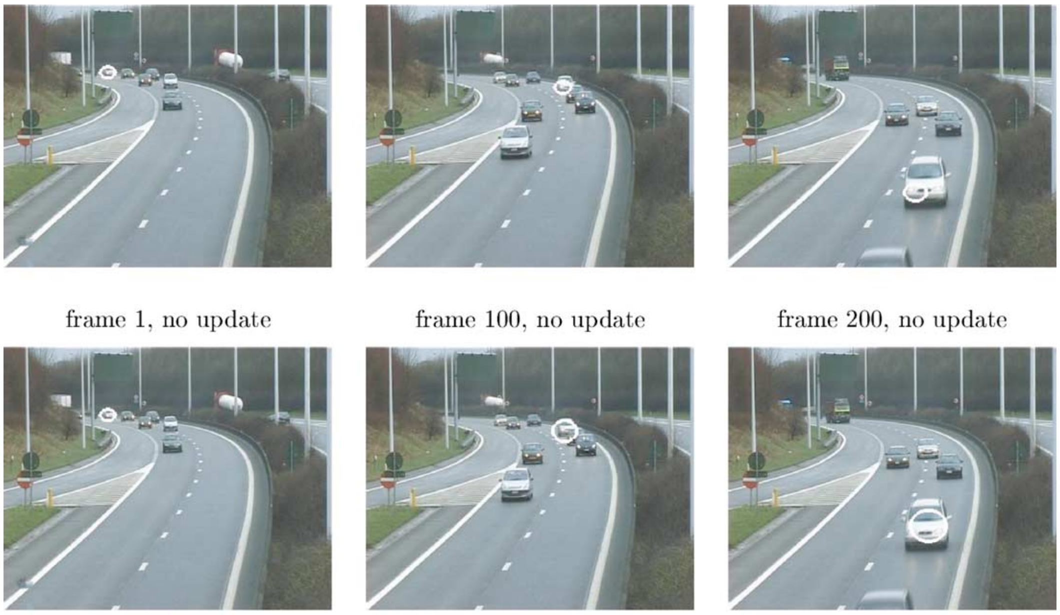
frame 1 frame 11 frame 19

frame 26 frame 31 frame 81

Fig. 14. The *moving stairs* sequence shows the robustness of the color-based particle filter against occlusion and strong scale changes. Furthermore, in frame 26 the effect of the initialization is illustrated.

appearance changes. Moreover, as multiple hypotheses are processed, objects can be tracked well in cases of occlusions or clutter. The proposed algorithm runs comfortably in real time with 10–30 frames per second without any special optimization on a normal 800 MHz PC.

The object model is represented by a weighted histogram which takes into account both the color and the shape of the target. The number of bins in the histogram should be optimized with respect to the noise of the camera, as too many bins can otherwise pose a problem. In these cases, a different similarity measure could be considered that also takes into account neighboring bins. In addition, further improvements can be achieved by using a different weighting function for



frame 1, no update frame 100, no update frame 200, no update

frame 1, with update frame 100, with update frame 200, with update

Fig. 15. The *traffic* sequence illustrates the importance of an adaptive target model in cases of occlusions and large scale changes. The white ellipses represent the mean states of the underlying sample distribution of N=100 elements. In the top row tracking without a model update is seen to result in a failure of scale adaptations, while in the bottom row a model update is applied and the scale remains correct.

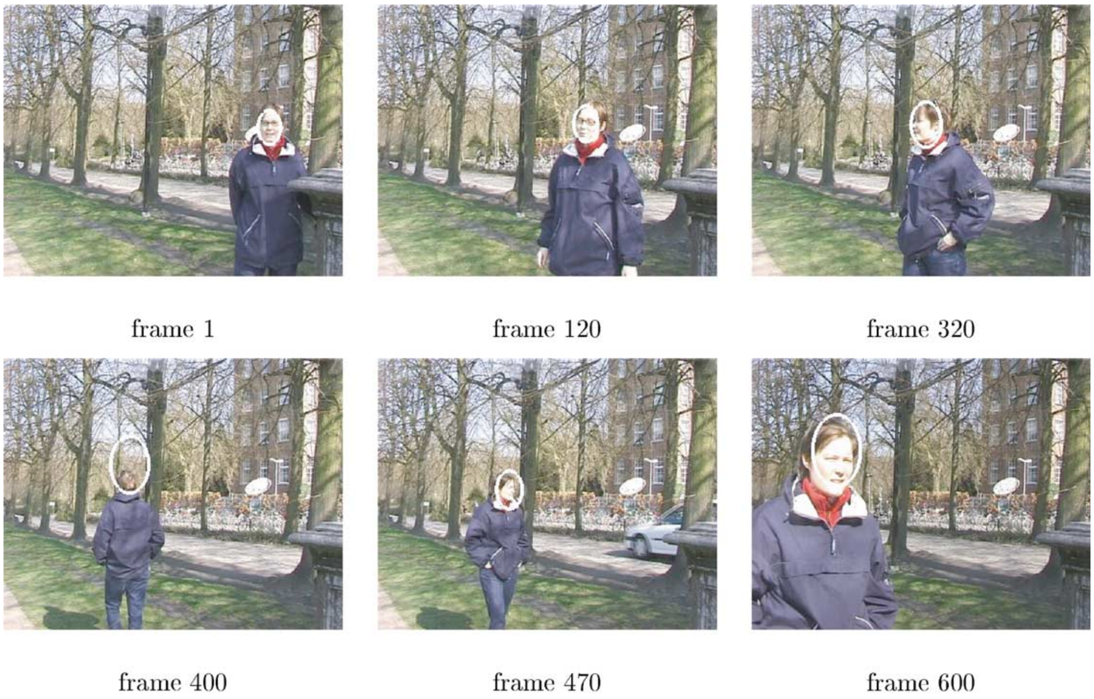


Fig. 16. The *face* sequence shows the tracking performance with an adaptive model. The tracker handles a large object motion and illumination changes using $N = 100$ samples.

the histograms to put more emphasis on the shape of the object, i.e. to utilize some prior knowledge of the expected object silhouette to calculate the weighted histogram.

A straightforward kinematic system model is currently used to propagate the sample set. By incorporating more a priori knowledge, for example by employing a learned motion model, the quality of the tracking could be further improved. The application of an adaptive model always implies a trade-off between an increasing sensitivity to extended occlusions and a more reliable tracking under appearance changes.

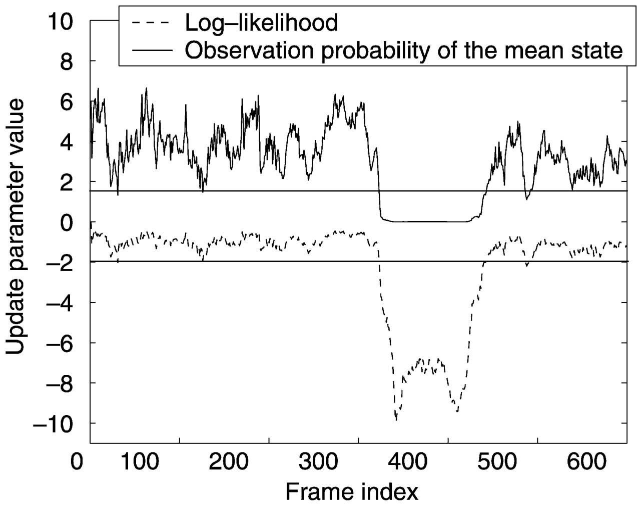


Fig. 17. The log-likelihood L and the observation probability $\pi_{E[S]}$ (here scaled with factor 10) can be both applied as an update rule with an appropriate threshold.

Our research interests now focus on multiple camera systems that can exchange information about the state of the objects that they track.

Acknowledgements

The authors acknowledge the support by the European IST project STAR (IST-2000-28764) and by the GOA/VHS + project financed by the Research Fund of Katholieke Universiteit Leuven, Belgium.

References

[1] F. Aherne, N. Thacker, P. Rockett, The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data, Kybernetika 32 (4) (1997) 1–7.

[2] M. Black, A. Jepson, A Probabilistic Framework for Matching Temporal Trajectories: Condensation-Based Recognition of Gestures and Expressions, European Conference on Computer Vision (1998) 909–924.

[3] D. Beymer, P. McLauchlan, B. Coifman, J. Malik, A Real-time Computer Vision System for Measuring Traffic Parameters, Computer Vision and Pattern Recognition (1997) 495–501.

[4] C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery (1998) 121–167.

[5] D. Comaniciu, V. Ramesh, P. Meer, Real-Time Tracking of Non-Rigid Objects using Mean Shift, Computer Vision and Pattern Recognition (2000) 142–149.

[6] D. Comaniciu, V. Ramesh, Mean Shift and Optimal Prediction for Efficient Object Tracking, International Conference on Image Processing (2000) 70–73.

[7] N. Friedman, S. Russell, Image Segmentation in Video Sequences: A Probabilistic Approach, Uncertainty in Artificial Intelligence (1997) 175–181.

[8] N. Gordon, D. Salmond, Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter, Journal of Guidance, Control and Dynamics 18 (6) (1995) 1434–1443.

[9] M. Greiffenhagen, V. Ramesh, D. Comaniciu, H. Niemann, Statistical Modeling and Performance Characterization of a Real-Time Dual Camera Surveillance System, Computer Vision and Pattern Recognition (2000) 335–342.

[10] G. Halevy, D. Weinshall, Motion of Disturbances: Detection and Tracking of Multi-Body Non-Rigid Motion, Machine Vision and Applications 11 (1999) 122–137.

[11] T. Heap, D. Hogg, Wormholes in Shape Space: Tracking through Discontinuous Changes in Shape, International Conference on Computer Vision (1998) 344–349.

[12] M. Isard, A. Blake, Contour Tracking by Stochastic Propagation of Conditional Density, European Conference on Computer Vision (1996) 343–356.

[13] M. Isard, A. Blake, A Mixed-State Condensation Tracker with Automatic Model-Switching, International Conference on Computer Vision (1998) 107–112.

[14] M. Isard, A. Blake, CONDENSATION – Conditional Density Propagation for Visual Tracking, International Journal on Computer Vision 1 (29) (1998) 5–28.

[15] M. Isard, J. MacCormick, BraMBLe: A Bayesian Multiple-Blob Tracker, International Conference on Computer Vision (2001) 34–41.

[16] A. Jepson, D. Fleet, T. El-Maraghi, Robust Online Appearance Models for Visual Tracking, Computer Vision and Pattern Recognition (2001) 415–422.

[17] T. Kailath, The Divergence and Bhattacharyya Distance Measures in Signal Selection, IEEE Transactions on Communication Technology COM-15 (1) (1967) 52–60.

[18] G. Kitagawa, Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, Journal of Computational and Graphical Statistics 5 (1) (1996) 1–25.

[19] D. Koller, J. Weber, J. Malik, Robust Multiple Car Tracking with Occlusion Reasoning, European Conference on Computer Vision (1994) 189–196.

[20] J. MacCormick, A. Blake, A Probabilistic Exclusion Principle for Tracking Multiple Objects, International Conference on Computer Vision (1999) 572–587.

[21] S. McKenna, Y. Raja, S. Gong, Tracking Colour Objects Using Adaptive Mixture Models, Image and Vision Computing 17 (1999) 225–231.

[22] B. Menser, M. Brünig, Face Detection and Tracking for Video Coding Applications, Asilomar Conference on Signals, Systems, and Computers (2000) 49–53.

[23] K. Nummiaro, E. Koller-Meier, L. Van Gool, A Color-Based Particle Filter, First International Workshop on Generative-Model-Based Vision, in conjunction with ECCV'02 (2002) 53–60.

[24] K. Nummiaro, E. Koller-Meier, L. Van Gool, Object Tracking with an Adaptive Color-Based Particle Filter, Symposium for Pattern Recognition of the DAGM (2002) 353–360.

[25] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-Based Probabilistic Tracking, European Conference on Computer Vision (2002) 661–675.

[26] Y. Raja, S. McKenna, S. Gong, Tracking and Segmenting People in Varying Lighting Conditions using Colour, International Conference on Face and Gesture Recognition (1998) 228–233.

[27] J. Segen, S. Pingali, A Camera-Based System for Tracking People in Real Time, International Conference on Pattern Recognition (1996) 63–67.

[28] A. Yilmaz, K. Shafique, N. Lobo, X. Lin, T. Olson, M. Shah, Target-Tracking in FLIR Imagery Using Mean-Shift and Global Motion Compensation, Computer Vision Beyond the Visible Spectrum (2001) 54–58.