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 $\pi_{t-1}^{\langle n\rangle}$:
  (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.

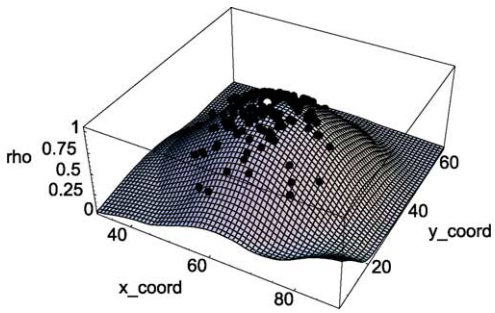

Fig. 3. Surface plot of the Bhattacharyya coefficient of a small area around the face of the soccer player shown in Fig. 1. The black points illustrate the centers of the ellipses of the sample set while the white point represents the mean location. It is positioned close to the maximum of the plot.

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 $q = \{q^{(u)}\}_{u=1\ldots m}$ 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, $\pm 5\%$ 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 $8 \times 8 \times 8$ 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

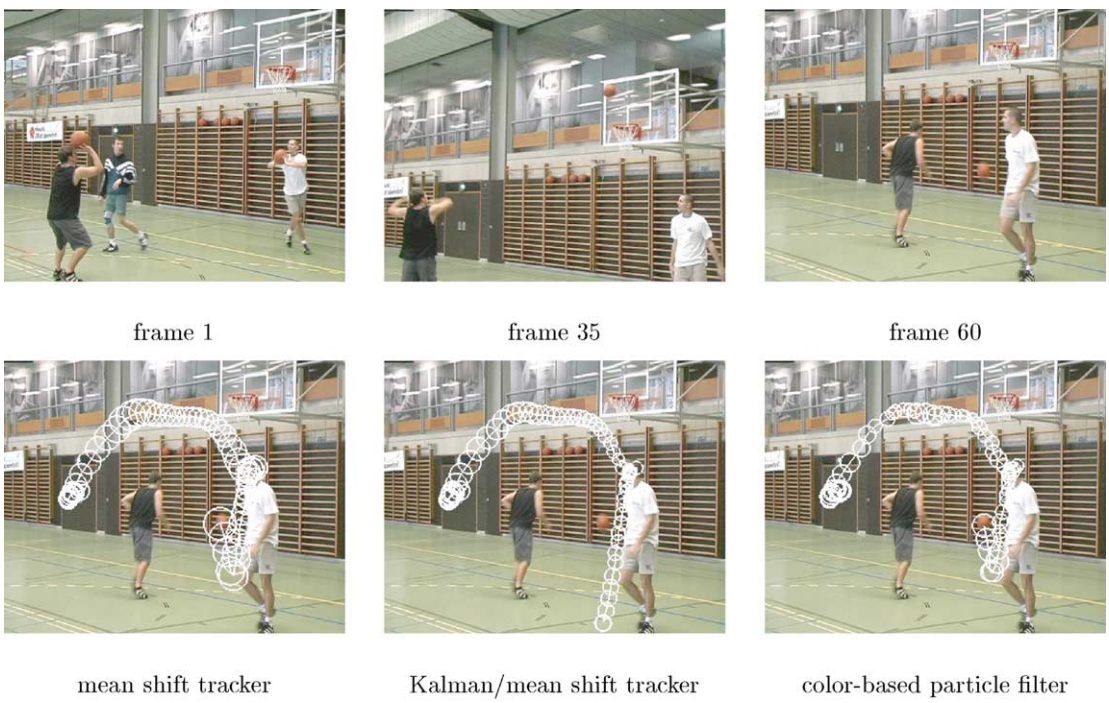

Fig. 5. Top row: The sequence of the ball thrown into the basket; Bottom row: The results of the mean shift tracker, the Kalman/mean shift tracker and the color-based particle filter.

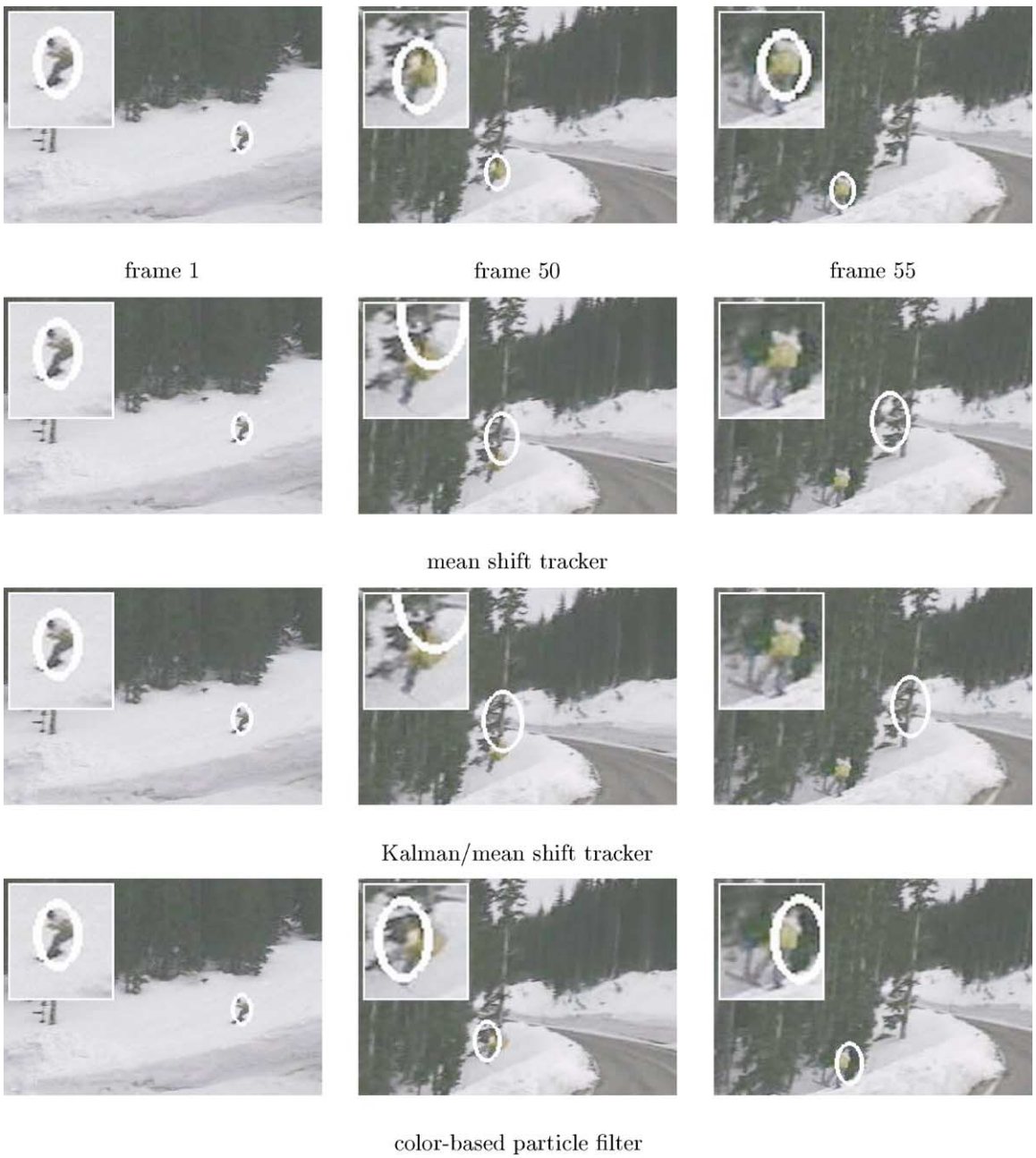

Fig. 6. The top row shows the sequence of a snowboarder jumping. The second row displays the results of the mean shift tracker, the third row shows the output of the combination of the mean shift tracker with the Kalman filter and in the last row the results of the color-based particle filter are shown.

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

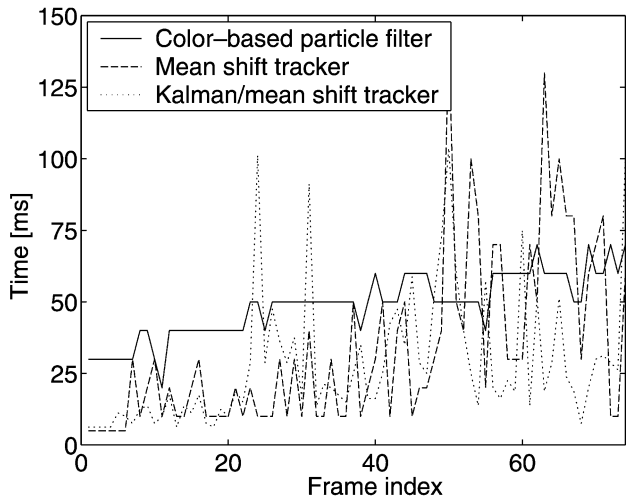

Fig. 7. Running times between two successive frames of the *snowboarder* sequence.

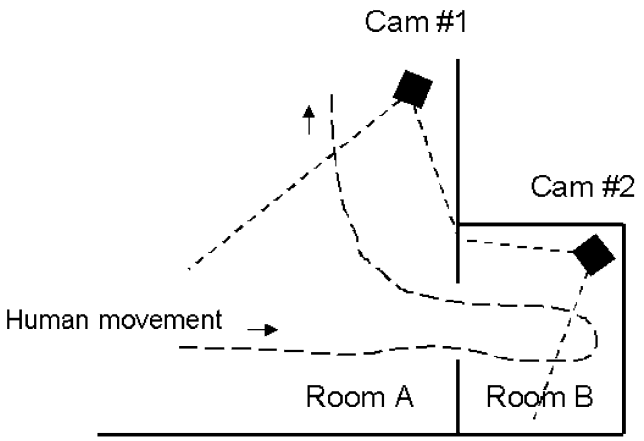

Fig. 8. Camera setup and the person's path in the *surveillance* example.

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.

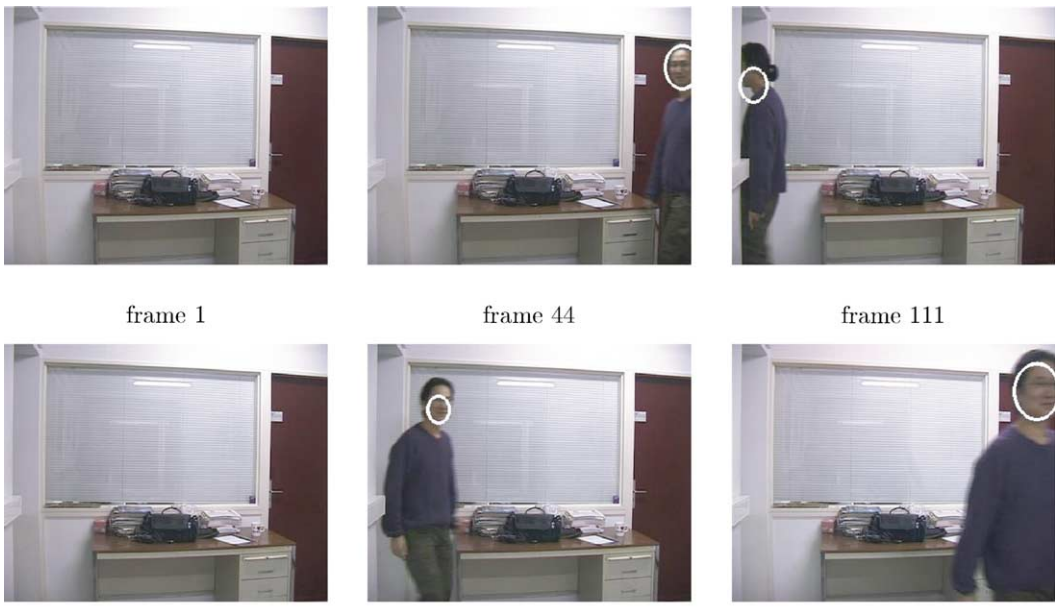

Fig. 9. Frame selection from the *surveillance* sequence of camera 1.

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.

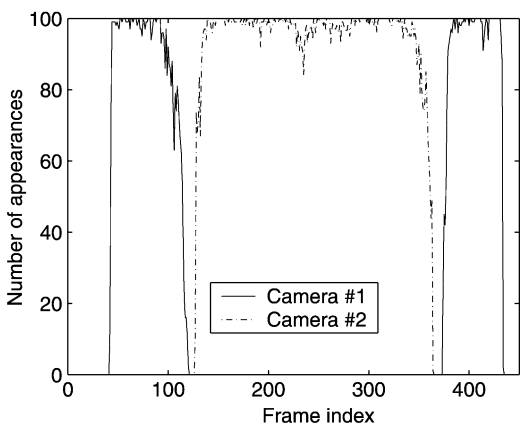

Fig. 11. The number of samples in each frame that fulfill the appearance condition for the *surveillance* sequence.

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

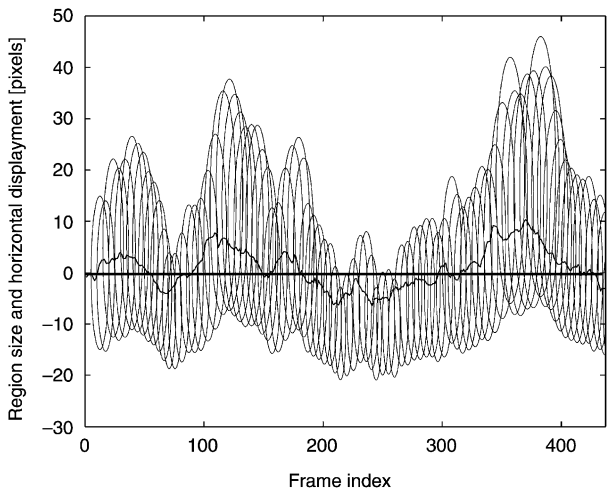

Fig. 13. The size and horizontal position of the elliptic object region during the *soccer* sequence is shown for every fifth frame. The scaling changes relatively smoothly. The line indicates the horizontal displacement which is relatively small as the player stays more or less in the center of the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

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

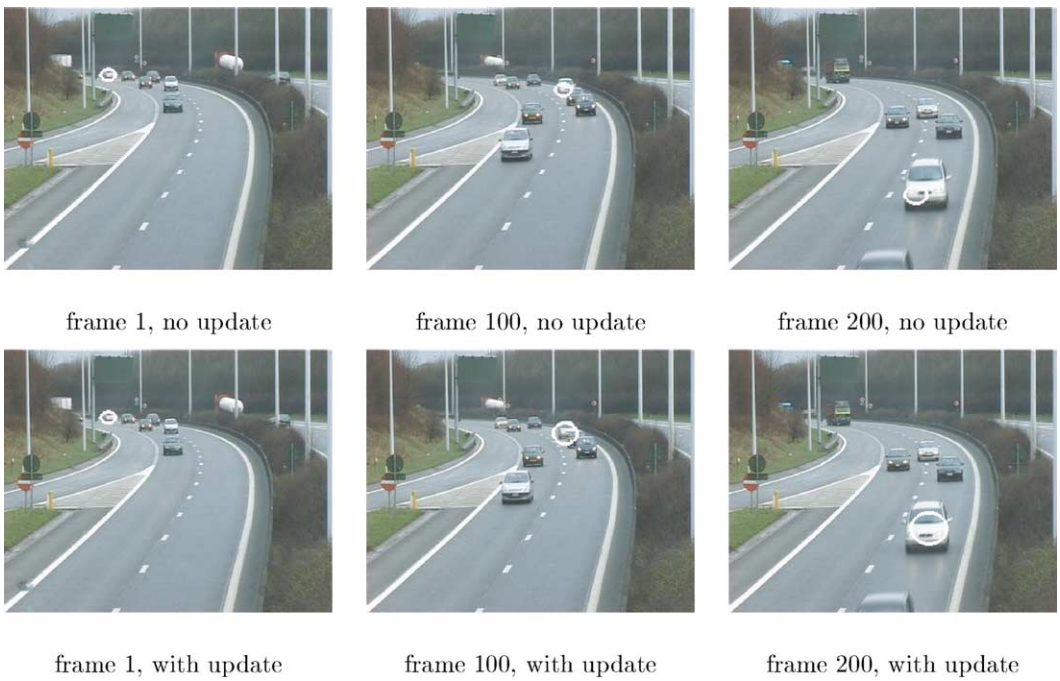

Fig. 15. The *traffic* sequence illustrates the importance of an adaptive target model in cases of occlusions and large scale changes. The white ellipses represent the mean states of the underlying sample distribution of N=100 elements. In the top row tracking without a model update is seen to result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

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

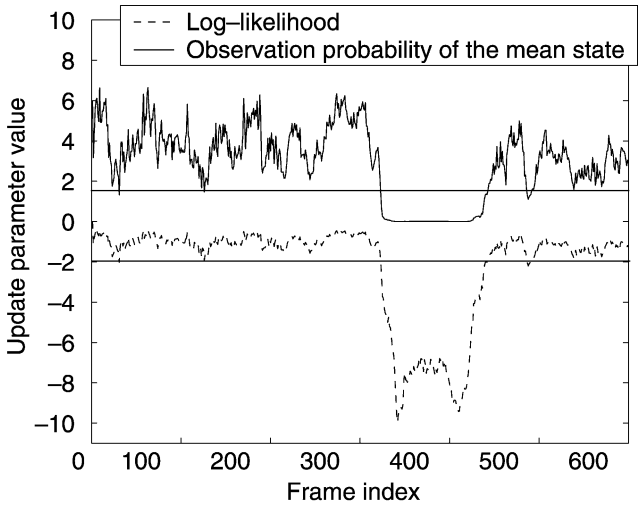

Fig. 17. The log-likelihood L and the observation probability $\pi_{E[S]}$ (here scaled with factor 10) can be both applied as an update rule with an appropriate threshold.

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.