A Color-based Particle Filter

Katja Nummiaro[1], Esther Koller-Meier[2] and Luc Van Gool[1,2]
[1] Katholieke Universiteit Leuven, ESAT/PSI Visics, Belgium
[2] Swiss Federal Institute of Technology (ETH), D-ITET/BIWI, Switzerland

Abstract— **Robust real-time tracking of non-rigid objects is a challenging task. Particle filtering has been proven very successful for non-linear and non-Gaussian estimation problems. However, for the tracking of non-rigid objects, the selection of reliable image features is also essential.**

This paper presents the integration of color distributions into particle filtering, which has typically used edge-based image features. Color distributions are applied as they are robust to partial occlusion, are rotation and scale invariant and computationally efficient. Thus, the target model of the particle filter is defined by the color information of the tracked object. As the tracker should find the most probable sample distribution, the model is compared with the current hypotheses of the particle filter using the Bhattacharyya coefficient, which is a popular similarity measure between two distributions. The proposed tracking method directly incorporates the scale and motion changes of the objects. Comparisons with the well known mean shift tracker show the advantages and limitations of the new approach.

Keywords— **Object tracking, Condensation algorithm, Color filtering, Bhattacharyya coefficient, Mean shift tracking**

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

I. Introduction

Object tracking is required by many vision applications such as human-computer interfaces [1], video communication/compression [2] or surveillance [3], [4], [5]. 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. They have typically been used with edge-based image features [6], [7], [8]. The idea of a particle filter – to apply a recursive Bayesian filter based on sample sets – was independently proposed by several research groups [7], [9], [10]. Our work has evolved from the Condensation algorithm [7] which was developed in the computer vision community. At the same time this filtering method was studied both for statistics and signal processing known as Bayesian bootstrap filter [9] or Monte Carlo Filter [10].

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 model is tracked with a particle filter by comparing its histogram with the histograms of the sample positions using the Bhattacharyya distance. A complete segmentation of the image is not required as the image content only needs to be evaluated at the sample positions. Figure 1 shows an application of the color-based particle filter for tracking the face of a soccer player.

A related approach is the mean shift tracker [11] which 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. In comparison to edge-based particle filters [6], [7], [8], our target model is more robust against out-of-plane rotations. In [12] color information has been used in particle filtering for initialization and importance sampling. Furthermore, in a recent paper [13], color cues have been employed in a foreground and background model using Gaussian mixtures. 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.

The outline of this paper is as follows. In Section II we briefly describe particle filtering and in Section III we indicate how color distributions are used as object models. The integration of the color information into the particle filter is explained in Section IV. As tracked objects may disappear and reappear an initialization based on an appearance rule is introduced in Section V. Section VI compares the mean shift tracker [11] with our proposed tracking framework as both methods use related color features as object models. In Section VII we then present some experimental results for surveillance and soccer scenes.

E-mail: knummiar@esat.kuleuven.ac.be .

II. Particle Filtering

Particle filtering [7] was developed to track objects in clutter, in which the posterior density $p(X_t|Z_t)$ and the observation density $p(Z_t|X_t)$ are often non-Gaussian. The quantities of a tracked object are described in the state vector X_t while the vector Z_t denotes all the observations $\{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ up to time t.

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 consists of an element $\mathbf{s}$ which represents the hypothetical state of an object and 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. Then, each element of the set is 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.

III. Color Distribution Model

We want to apply such a particle filter in a color-based context. To achieve robustness against non-rigidity, rotation and partial occlusion we focus on color distributions as target models. These are represented by histograms which are produced with the function $h(\mathbf{x}_i)$, that assigns one of the m-bins to a given color at location $\mathbf{x}_i$. The histograms are typically calculated in the RGB space using 8x8x8 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. 8x8x4 bins).

Not all pixels in the region are equally important to describe the objects. For example, pixels that are further away from the region center can be assigned smaller weights by employing a weighting function

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

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 [11].

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 δ is the Kronecker delta function and the parameter a is used to adapt the size of the region. 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 [14], [15]

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

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 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.

IV. Color-based Particle Filtering

The proposed tracker employs the Bhattacharyya distance to update the a priori distribution calculated by the particle filter. The target regions are represented by ellipses, so that a sample is given as

$$\mathbf{s} = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{H}_x, \dot{H}_y\} \tag{8}$$

where x, y represent the location of the ellipse, $\dot{x}$, $\dot{y}$ the motion, H_x, H_y the length of the half axes and $\dot{H}_x$, $\dot{H}_y$ the corresponding changes. 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

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

where A defines the deterministic and $\mathbf{w}_{t-1}$ the stochastic component. In our application we currently use a first order model for A describing an object moving with constant velocity for x, y, H_x and H_y. Expanding this model to second order is straightforward.

To weigh the sample set, the Bhattacharyya coefficient has to be computed between the target distribution and the distribution of the hypotheses. Each hypothetical region is specified by its state vector $\mathbf{s}^{(n)}$. Both the target q and the candidate histogram $p(\mathbf{s}^{(n)})$ are calculated from Eq. 3 where the target is centered in the origin and $a = \sqrt{H_x^2 + H_y^2}$. Currently a fixed target model is used, but an adaptive approach as presented in [16], [17] could also be implemented.

As we want to favor samples whose color distributions are similar to the target model, the Bhattacharyya distance is used for the weighting. The probability of each sample

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

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}
   $c_{t-1}^{(0)} = 0$
   $c_{t-1}^{(n)} = c_{t-1}^{(n-1)} + \pi_{t-1}^{(n)}$
   $c'^{(n)}_{t-1} = \frac{c_{t-1}^{(n)}}{c_{t-1}^{(N)}}$
   (b) generate a uniformly distributed random number $r \in [0,1]$
   (c) find, by binary search, the smallest j for which $c'^{(j)}_{t-1} \geq r$
   (d) set $\mathbf{s}'^{(n)}_{t-1} = \mathbf{s}^{(j)}_{t-1}$
2. **Propagate** each sample from the set S'_{t-1} by a linear stochastic differential equation:
   $\mathbf{s}_t^{(n)} = A\,\mathbf{s}'^{(n)}_{t-1} + \mathbf{w}_{t-1}^{(n)}$
   where $\mathbf{w}_{t-1}^{(n)}$ is the stochastic component
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
   (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
   $\pi_t^{(n)} = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{(1-\rho[p(\mathbf{s}_t^{(n)}), q])}{2\,\sigma^2}}$
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.

is 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 Figure 2.

To illustrate the distribution of the sample set, Figure 3 shows the Bhattacharyya coefficient of a small area around the tracked face of the soccer player shown in Figure 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 maxima and consequently the localization of the face is very accurate.

V. 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

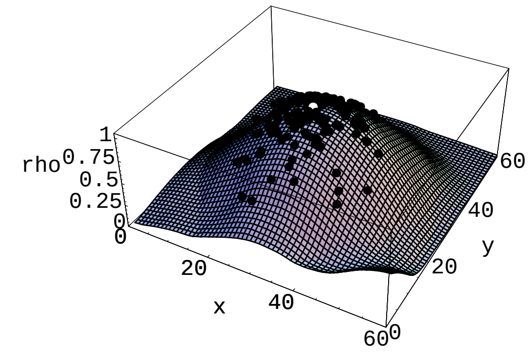

Fig. 3. Surface plot of the Bhattacharyya coefficient of a small area around the face of the soccer player shown in Figure 1. The black points illustrate the centers of the ellipses of the sample set while the white point represents the mean location.

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 recognition algorithm that finds interesting targets. The manual initialization requires operator interaction at the beginning of the tracking. Furthermore, the whole object must be fully visible, so that a good color distribution can be calculated.

If the object histogram $q = \{q^{(u)}\}_{u=1\ldots m}$ is known, we can place samples strategically at positions where the target is expected to appear (see 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 every pixel of a background image:

$$\mu \quad = \quad \frac{1}{M} \sum_{i=1}^{M} \rho[p(\mathbf{x}_i), q] \tag{11}$$

Fig. 4. The samples are spread over the image areas, where the known object is most likely to appear, like doors and image borders.

frame 1

frame 50

frame 55

Fig. 5. *Snowboarder* sequence tracked with the mean shift algorithm. In frame 50 the tracker loses the object and it is not able to recover afterwards.

frame 1

frame 50

frame 55

Fig. 6. *Snowboarder* sequence tracked with the color-based particle filter. The tracker is able to follow the fast moving object during the whole sequence.

$$\sigma^2 \quad = \quad \frac{1}{M} \sum_{i=1}^{M} (\rho[p(\mathbf{x}_i), q] - \mu)^2 \tag{12}$$

and then define an appearance rule as

$$\rho[p(\mathbf{s}_t^{(n)}), q] > \mu + 2\sigma. \tag{13}$$

This indicates a 95% confidence that a sample does not belong to the background. If more than a fraction f of the sample set fulfills the appearance rule during initialization, we consider the object to be found and start tracking.

Likewise, the same rule is used to determine if an object is lost during the tracking. If the number of positive appearances is smaller than f for a couple of frames, the tracker enters the 'initialization' mode. In our experiments a value of $f = 0.1$ has been proven sufficient.

As the most advanced initialization method we use a face detection scheme based on Support Vector Machines [18] to find a target location and calculate the color distribution to initialize the particle filter. Using existing collections of classes *faces* and *non-faces*, the data classification algorithm based on SVM calculates an optimal hyper-plane that separates these two classes. After training, an image region is classified to determine whether it contains a face or not. When a face has been recognized, the particle filter is initialized at the found position.

VI. Comparison with the Mean Shift Tracker

The mean shift algorithm has been introduced recently in several computer vision papers for tracking and segmentation applications [11], [19]. It is a simple and fast adaptive tracking procedure that finds the maximum of the Bhattacharyya coefficient given a target model and the starting location x, y and size H_x, H_y. Based on the mean shift vector which is an estimation of the gradient, the new object location is calculated. This step is repeated until the target location no longer significantly changes. The 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 illustrate the differences between the mean shift tracker and our proposal we consider the *snowboarder* se-

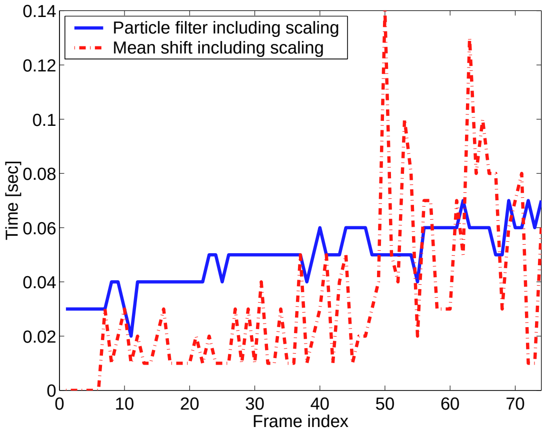

Fig. 7. Running times between two successive frames of the *snowboarder* sequence.

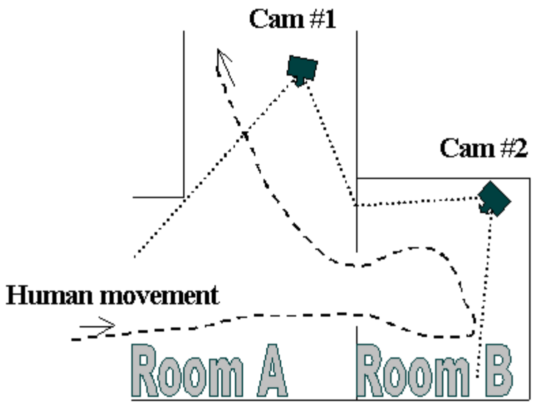

Fig. 8. Camera setup and the human movement of the *surveillance* example.

quence which is processed with a 800Mhz Pentium3 PC under Linux. The goal is to follow the snowboarder during his jump. The image size is 352x240 pixels and the dimensions of the initial elliptical search region is 14x22 pixels. The sequence has been processed using the two trackers. Figure 5 shows the result of the mean shift algorithm and Figure 6 the result of our color-based particle filter using $N = 100$ samples. In both cases scaling was included and the histograms were calculated in the RGB color space using 8x8x8 bins.

One of the biggest problems in motion-based tracking is to lose the object due to rapid movements. Such a situation is shown in Figure 5 on frame 50. If the old and new region do not overlap the mean shift tracker can converge to a local maxima on 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 starting location for the mean shift iterations [20], the situation improves as the regions no longer need to overlap and the tracker is more likely to converge to the correct maxima. However, as a single hypothesis is used the tracker will still have problems to follow the object through clutter. In contrast, the particle filter tracks multiple hypotheses and is therefore more reliable. For a more detailed discussion about the advantages of particle filtering vs. Kalman filtering in regard to non-linear and non-Gaussian problems see [7].

The mean shift algorithm has itself no scale adaptation. In practice, this has to be handled by calculating the Bhattacharyya distance with different fixed scale values in order to detect possible size changes during the sequence. In our particle filtering the scale is directly integrated and propagated using the system model. Consequently, the scale changes unrestricted and adapts better to the actual size of the object and is more accurate. As a further improvement the acceleration of the movement could also be included into the state vector. This facilitates to track objects with more complex movements with rapid stops and turns.

The mean shift tracker has the advantage that a more precise localization is calculated that corresponds to a maxima of the similarity. In particle filtering the mean value of the sample distribution has to be calculated as an estimation of the object location. 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 both approaches as several color distributions have to be calculated. When using mean shift, the number of these calculations depends on the number of iterations, and in particle filtering on the number of samples. The computing times of both trackers for the *snowboarder* sequence are shown in Figure 7. On the average, the mean shift tracker is faster but needs more computation time in frames where it is losing the object. However, both trackers run in real time.

VII. Results

Some of the most important object tracking applications are surveillance systems. These kind of systems must perform many computer vision tasks like object tracking, recognition, classification and analysis of different actions.

We consider such a *surveillance* sequence of 450 frames to demonstrate the efficiency of our approach. The system uses two fixed cameras to track a person who is moving inside two connected rooms. The configuration is shown in Figure 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 closed room B that is observed by camera 2. Currently, the trackers in both cameras are working independently, i.e. each of them uses a separate particle filter.

For our surveillance task of following a person inside the rooms, we use a first order kinematic model for the system dynamics (Eq. 9). The new state parameters are predicted based on the old values and an uncertainty vector. The noise terms are chosen proportional to the size of the initial region. In fact, both the system and noise models represent prior knowledge of the tracking framework.

In this experiment we used the initialization method based on a known histogram. Figure 9 shows the target histogram of the face/head tracker. Both trackers are put

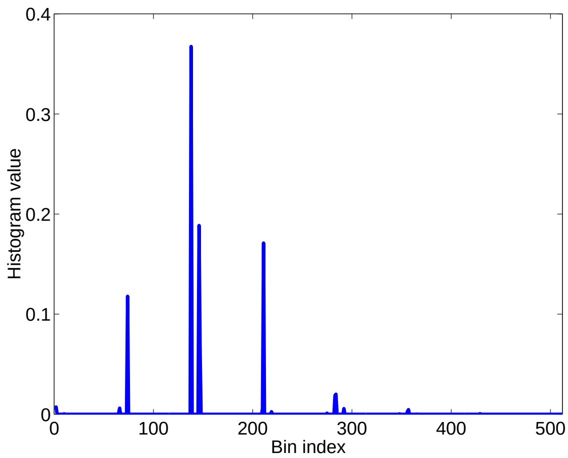

Fig. 9. Known face histogram for the *surveillance* example.

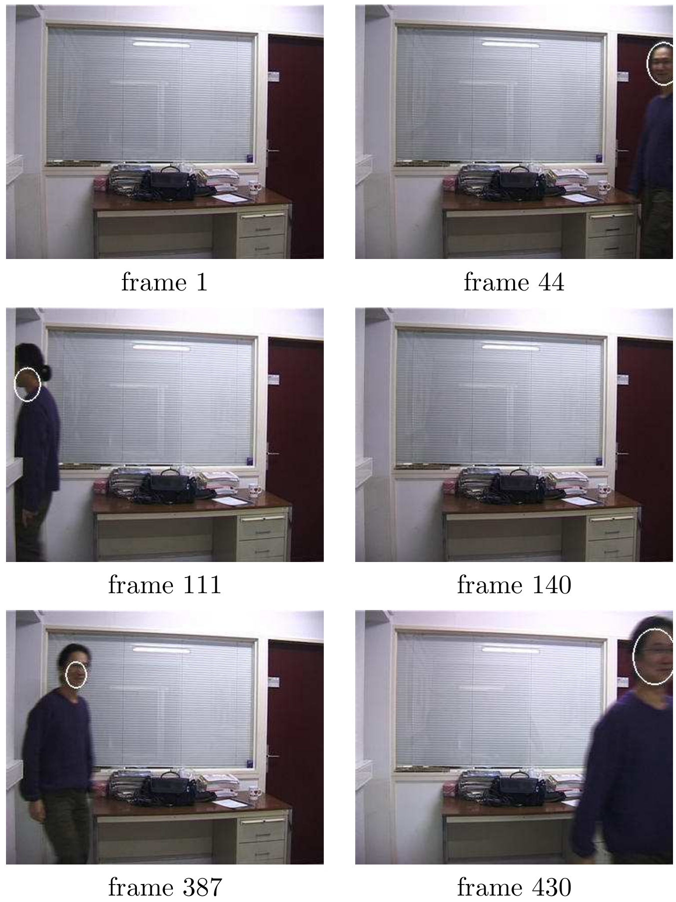

Fig. 10. Frame selection from the *surveillance* sequence of camera 1.

into the 'initialization' mode and will 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 Figure 10 and Figure 11 the results of camera 1 respectively 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 processed. 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 given 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 rule given in Eq. 13. Figure 12 shows the number of positive appearances for both cameras. In this experiment we used $N = 100$ samples and the fraction was set to $f = 0.1$.

To demonstrate the robustness of our particle filter against occlusion and rapid movements, we applied it to a *soccer* sequence and let the tracker follow a single player over 438 frames. The results are displayed in Figure 13. In this sequence the camera is not fixed but 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. Figure 14 shows the sizes and displacements of the object region during the *soccer* sequence. It can be seen, that the scaling changes quite smoothly.

In Figure 15 we consider a *moving stairs* sequence of 90 frames in a train station. A static surveillance camera is installed to track the faces of passing passengers. In this experiment the efficiency of color-based particle filtering against occlusion and large scale changes as well as the effect of the 'initialization' mode can be observed. 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 rule given in Eq. 13. The new initial location is poor at the beginning but improves quickly after a few frames with the use of multiple hypotheses.

In the last experiment we want to point out a current limitation of the tracker. Figure 16 shows a *traffic* sequence of 257 frames recorded by a freeway monitoring system. As we are currently working with a fixed target model, the resulting region gets stuck on the left front side of the car. This is the case, as the target histogram was chosen in the first frame and not updated over the sequence. To overcome this problem an adaptive target model could be implemented similar as in [16], [17].

VIII. Conclusion

The proposed tracking method adds the robustness and invariance of color distributions to particle filtering. The tracker can successfully handle a fast moving object, even if its search regions do not overlap in consecutive frames. As multiple hypotheses are generated, objects can be well tracked in cases of occlusion or clutter. The proposed algorithm runs comfortably in real time with 10 to 30 frames

Fig. 11. Frame selection from the *surveillance* sequence of camera 2.

per second without any special optimization.

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 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 the histograms to put more emphasis on the shape of the object.

We are currently using a straight forward kinematic system model 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.

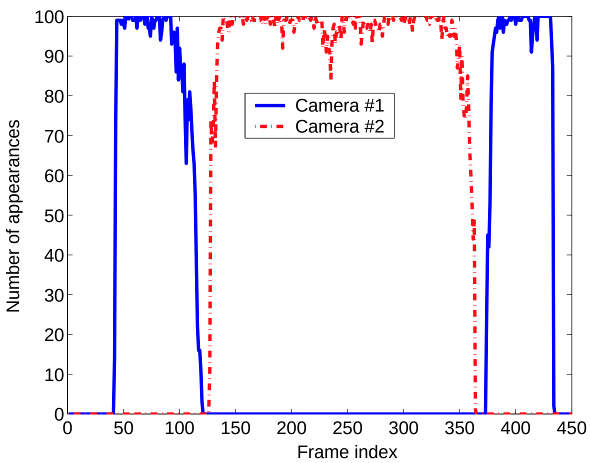

Fig. 12. The number of samples in each frame that fulfill the appearance rule for the *surveillance* sequence.

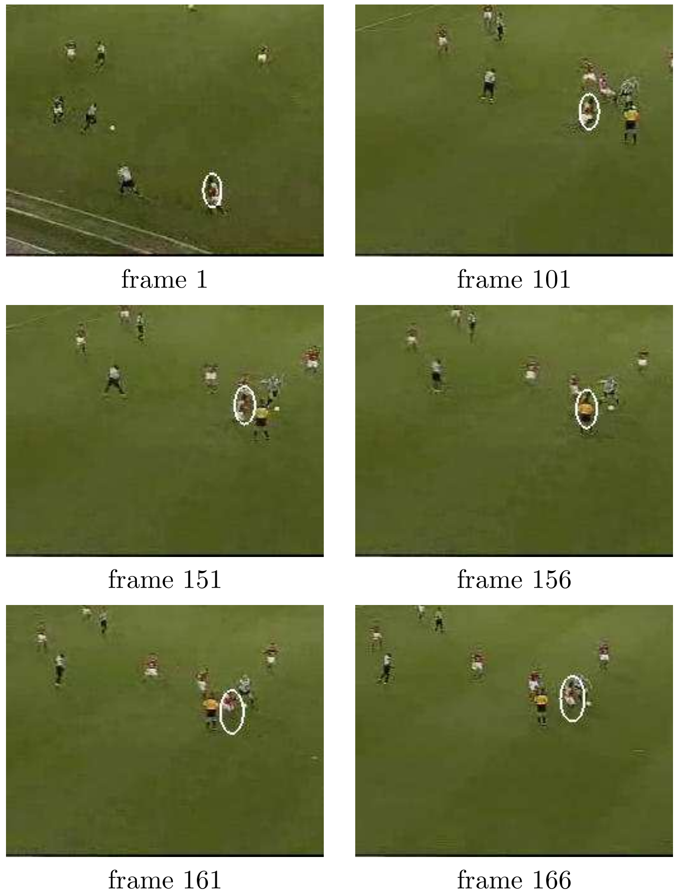

Fig. 13. This *soccer* sequence shows the successful tracking of a player in cases of occlusion and rapid movement.

The fixed target model poses a limitation in scenes where the object changes significantly. Our research interests now focus on slowly adapting models which can cope with appearance changes. We are also investigating multiple camera systems that can exchange information about the state of the objects that they track.

Acknowledgment

This research was partially supported by STAR (IST-2000-28764) which is a project of the European Commission and by the GOA/99/05-VHS project financed by Research Fund K.U.Leuven. The authors would like to thank David Serby for providing the implementation of the particle filter.

References

[1] M. Black and A. Jepson, “A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions,” in *European Conference on Computer Vision*, 1998, vol. 1, pp. 909–924.

[2] B. Menser and M. Brünig, “Face detection and tracking for video coding applications,” in *Asilomar Conference on Signals, Systems, and Computers*, 2000, vol. 1, pp. 49–53.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “A real-time computer vision system for measuring traffic parameters,” in *Computer Vision and Pattern Recognition*, 1997, pp. 495–501.

[4] M. Greiffenhagen, V. Ramesh, D. Comaniciu, and H. Niemann, “Statistical modeling and performance characterization of a real-

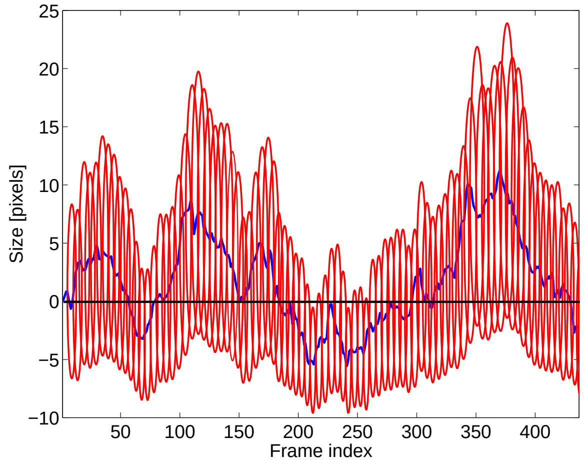

Fig. 14. The sizes and displacements along the x-coordinate of the object region during the *soccer* sequence is shown for every fifth frame. The scaling changes relatively smoothly.

time dual camera surveillance system," in *Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 335–342.
[5] J. Segen and S. Pingali, "A camera-based system for tracking people in real time," in *International Conference on Pattern Recognition*, 1996, vol. 3, pp. 63–67.
[6] T. Heap and D. Hogg, "Wormholes in shape space: Tracking through discontinuous changes in shape," in *International Conference on Computer Vision*, 1998, pp. 344–349.
[7] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *International Journal on Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
[8] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," in *International Conference on Computer Vision*, 1999, vol. 1, pp. 572–587.
[9] N. Gordon and D. Salmond, "Bayesian state estimation for tracking and guidance using the bootstrap filter," *Journal of Guidance, Control and Dynamics*, vol. 18, no. 6, pp. 1434–1443, 1995.
[10] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.
[11] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 142–149.
[12] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," in *European Conference on Computer Vision*, 1998, vol. 1, pp. 893–908.
[13] M. Isard and J. MacCormick, "BraMBLe: A bayesian multiple-blob tracker," in *International Conference on Computer Vision*, 2001, pp. 34–41.
[14] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 32, no. 4, pp. 1–7, 1997.
[15] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. COM-15, no. 1, pp. 52–60, 1967.
[16] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 415–422, 2001.
[17] Y. Raja, S. McKenna, and S. Gong, "Tracking and segmenting people in varying lighting conditions using colour," in *3rd International Conference on Face and Gesture Recognition*, 1998, pp. 228–233.
[18] C. Burges, "A tutorial on support vector machines for pattern recognition," vol. 2, pp. 121–167, 1998.
[19] A. Yilmaz, K. Shafique, N. Lobo, X. Lin, T. Olson, and M. Shah, "Target-tracking in FLIR imagery using mean-shift and global motion compensation," in *Computer Vision Beyond the Visible Spectrum*, 2001, pp. 54–58.
[20] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *International Conference on Image Processing*, 2000, vol. 3, pp. 70–73.

Fig. 15. The *moving stairs* sequence shows the efficiency of the color-based particle filter against occlusion and strong scale changes. Furthermore, in frame 26 the effect of the initialization is illustrated.

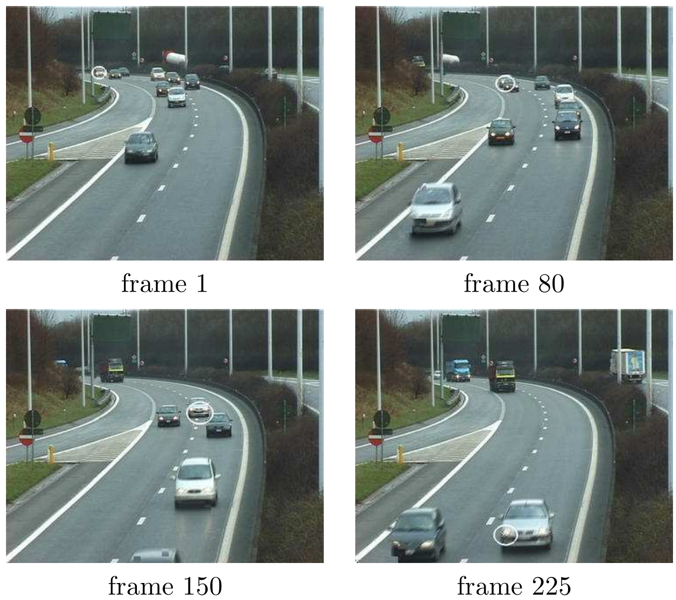

Fig. 16. *Traffic* sequence where the resulting region is inaccurate located on the left front side of the car due to large histogram changes of the target.