Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation

Alberto Del Bimbo, Fabrizio Dini

Abstract

In many real world applications, tracking must be performed reliably in real-time for sufficiently long periods where target appearance and motion may sensibly change from one frame to the following. In such non ideal conditions this is likely to determine inaccurate estimates of the target location unless dynamic components are incorporated in the model. To deal with these problems effectively, we propose a particle filter-based tracker that exploits a first order dynamic model and continuously performs adaptation of model noise to balance uncertainty between the static and dynamic components of the state vector. We provide an extensive set of experimental evidences with a comparative performance analysis with tracking methods representative of the principal approaches. Results show that the method proposed is particularly effective for real-time tracking over long video sequences with occlusions and erratic, non-linear target motion.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Visual tracking is the problem of consistently labeling a target object in the images of a video sequence. It plays a critical role in several applications, particularly in real world operation monitoring, video surveillance [1,2] and human–computer interaction [3–5], among the others [6–9].

However, despite of the research efforts and attention in the last decade, visual tracking still remains a challenging problem. Small target size, background clutter, low contrast with the background, appearance changes and erratic motion are some of the most common problems that make reliable tracking difficult to achieve. Moreover, in real cases, tracking must deal with total or partial occlusion of the target. In this case, the object being tracked can be lost for a few frames and should be recovered as soon as possible after the occlusion. In most cases it is also required that visual trackers are capable to provide reliable tracks over very long sequences and operate in real-time at high frame rates.

Visual tracking is strictly dependent on the effectiveness of target localization. This makes visual tracking and object detection two highly interrelated tasks. Indeed, object detectors can be employed effectively with a twofold function. On the one hand, they can detect new targets that enter in the scene and thus initialize trackers. On the other hand, they can provide accurate measures of the target position and thus help the tracker to provide accurate estimates. In this case, tracking can even be reduced to the task of matching the detections on the present frame with those in the previous frame (tracking-by-detection). Several works in the literature have followed tracking-by-detection to provide very accurate tracking [10–13].

Anyway, tracking-by-detection is usually restricted to the cases where the target appearance has some distinguishing characteristics, and has small changes in consecutive frames. It is not effective when no prior assumption can be made about target appearance, as for example with fast maneuvering targets when the appearance changes very quickly or with targets of small size and poor distinguishing features.

In this case, detection-by-tracking should be followed. In this approach, some low level weak appearance model is used and the probability density function (PDF) of the target position is estimated by sampling the image plane, and hence applying some transformation to the pixels to evaluate the similarity with the target. Solutions for detection-by-tracking fall into two major classes: deterministic tracking and stochastic tracking (for a complete classification, see [14]).

Mean-shift [15–17] and Continuously Adaptive Mean-shift (CAM-shift) [18] are the best known approaches in the class of deterministic trackers. For each frame, they define a cost function such that the target position corresponds to the minimum (or maximum) of the function value. In this way, visual tracking is reduced to the optimization of the cost function. Mean shift and CAM-shift use a gradient descent technique to iteratively browse the cost function.

Keywords:
Adaptive Particle Filter
Visual tracking
Uncertainty adaptation
First order dynamic model
function towards the minimum (or maximum). The choice of the starting point is obviously a critical factor in these trackers, which however are prone to the problem of local optima, that could eventually be far from the actual target position. Deterministic trackers are therefore inadequate for tracking in long video sequences. In fact, in this case, since the target appearance may drastically change over time, the target PDF is typically multimodal and target localization becomes extremely uncertain.

Stochastic trackers [19] are instead based on the Bayesian estimation theory [20, 21]. In this case, visual tracking is reduced to the problem of estimating the PDF of a state vector which describes the target’s position, based on iterative measures of the state vector value, taken at subsequent time steps. The problem with such measurement process is that it is sensitive to the noise that affects the measure of the state vector. Therefore, a filter is needed to recover the actual value of the state vector from these noisy measures. A well known representative of this class of trackers is the Kalman filter [22]. It is developed under the assumption that the state vector PDF is Gaussian, which allows an analytic solution for the PDF estimation to be computed. Unfortunately, this hypothesis does not always hold and moreover Kalman filter cannot be applied “as is” to non-linear systems. Modifications have been obtained by applying linearization (the Extended Kalman Filter [22]), or by deterministic sampling (the Unscented Kalman Filter [23, 24]). However, most of these solutions work satisfactorily only under very restrictive assumptions on the conditions of operation. Typically they rely on a number of parameters that must be adjusted to fit each specific problem domain. But a single set of parameters hardly succeeds in providing good results over long sequences or in situations with sudden changes in shape and appearance of the target.

Monte Carlo methods and particularly particle filter have gained interest for stochastic visual tracking for their intrinsic capability to adapt to changes, by tracking multiple hypotheses and coping with the non-linearity of the state and measurement processes, and for their simple and computationally efficient implementation [25, 26]. Although the true potential of particle filter for visual tracking remains largely not assessed due to the lack of systematic, large-scale empirical studies on its performance in critical conditions, limitations nevertheless should be expected when the target appearance is not strongly characterized (the filter ability to estimate a multimodal distribution may lead to noisy estimates of the target position) and when there are quick changes of target appearance and motion (in these cases the particles of the filter can be propagated in some wrong direction since the approximation of the target PDF given by the filter only holds in a bounded region of the state space, and the tracker can miss the target). Adaptation of particle filter-based tracking has therefore been proposed with reference to different aspects, in order to improve the performance of tracking in critical conditions. Particularly, target scale inference using particle filtering for adaptation of tracking to scale changes has been studied by several authors. In [27, 13], the target appearance is modeled as an elliptical shape and measured with a color histogram. The state vector includes the target position, scale and their derivatives so that the particle filter is able to predict the scale at which a target reappears after occlusions. In [28], the scale dynamics is not modeled in the target state, but is instead inferred from the estimated position, using the homographic transformation. The target dynamics is also obtained in 3D world coordinates. In [29], the authors have introduced an original solution to adapt the appearance model of the target, with application to the task of tracking human faces. In particular, faces are approximated by an elliptical region modeled using a wavelet-based appearance model. Tracking estimates the image motion and the appearance model, from the dominant image structure. Adaptation of the number of particles of the filter according to the variations of the target shape was instead proposed in [30]. In [31], Zhou et al. have defined a state vector that holds the six parameters of the affine transformation. The filter particles are distributed according to a random walk model. However, to account in some way for the target velocity, the state equation includes the appearance difference between the incoming observation and the previous particle configuration. Adaptation is performed over the number of particles based on the degree of uncertainty on the state value. They also use a mixture appearance model based on a modification of the Online Appearance Model that makes appearance adaptive and – as a consequence of it – the particle filter observation model adaptive as well. A similar solution was also followed in [32], where the authors also consider noise adaptation on position and scale of the target with different amounts from rotation and skew.

In this paper, we introduce a new method of adaptive particle filter-based tracking that propagates particles according to a first order dynamic model. The state of our generative model is a 8-dimensional vector including target position, width and aspect ratio of the target bounding box and their derivatives. Differently from other adaptive approaches, we do not apply adaptation to the number of filter particles nor to the target appearance. Instead, adaptation is applied to the standard deviations of the noise on the static and dynamic components of the state vector. Moreover, in order to avoid that the noise on the dynamic part amplifies the uncertainty on position and size of the target, constraints are applied on this component, so to control the propagation of uncertainty in the estimation. This solution is particularly suited for the cases where target appearance and motion change very much from one frame to the following as it likely occurs in long observation sequences with maneuvering targets and occlusions. We will prove that the solution proposed has high stability over long sequences, improves on the basic particle filter tracker while preserving the efficiency of the method (it performs approx 25 frames per second on a standard PC), and outperforms some of the most notable approaches to tracking. Since the solution proposed does not require any background modeling and only exploits the target appearance to track the target apparent motion in the sequence frames, it is also well suited to tracking with PTZ cameras provided that the target apparent maneuvers induced by the camera movements do not exceed the filter capability of tracking the target.

The presentation is organized as follows: in Section 2 we introduce tracking with particle filter with the explicit definition of both the state update and measurement equations; in Section 3 we present the solution to adaptively estimate the noise parameters. Finally, in Section 4 we assess the merits of the solution proposed with respect to the non-adaptive particle filter tracker, and perform an extensive comparison with several state of the art tracking methods, namely: the CAM-shift tracker (in the class of deterministic tracking), the adaptive approach by Zhou et al. [31] (in the class of detection-by-tracking, based on Adaptive Particle Filtering), and the method by Stalder et al. [12] (in the class of tracking-by-detection).

2. Tracking with the particle filter

Generally speaking, particle filter is based on a system of model and measurement time-dependent equations:

$$x_k = f_k(x_{k-1}, v_{k-1}).$$

$$z_k = h_k(x_k, n_k).$$

Eq. (1) is the system update equation, that represents the evolution of the state of the system from time $k-1$ to time $k$. The state $x_k$ depends in fact on the previous state $x_{k-1}$ of the system (we implicitly assume that the state update process admits a Markov property of order 1), and a stochastic error $v_{k-1}$ that represents
the uncertainty in the state update. Since $\mathbf{v}_{k-1}$ is a random variable of known statistics, the equation implicitly defines a probability density function $p(x_k|x_{k-1})$ (usually called prior PDF). Eq. (2) is the measurement equation, defining the dependency of the measure $z_k$ on the current unknown value of the state $x_k$ and the error term $n_k$ (representing the uncertainty in measuring the state). Since $n_k$ is a stochastic variable, also this equation implicitly defines a probability density function $p(z_k|x_k)$ that will be referred to in the following as measurement PDF (also known as likelihood).

2.1. State update equation

In our case, in order to provide a good compromise between the quality of target shape segmentation and the computational cost for real-time target tracking, we have approached the target shape with its rectangular bounding box and taken its color histogram for content representation. We have employed a first order motion model in order to predict the future target position in the next few frames of the sequence and avoid the exhaustive search over the entire frame. The state vector has therefore been defined so as to include the position and size of the rectangular bounding box of the target object in the image plane, as well as its velocity and changes in size. In particular, the state space for our tracker is defined over vectors of the form:

$$
\mathbf{x}_k = [x_k \ y_k \ w_k \ \rho_k \ \dot{x}_k \ \dot{y}_k \ \dot{w}_k \ \dot{\rho}_k]^T = [\mathbf{s}_k \ \mathbf{d}_k]^T.
$$

The first component $\mathbf{s}_k = [x_k \ y_k \ w_k \ \rho_k]$ specifies the static part of the state vector, and includes the coordinates $(x_k, y_k)$ of the upper left corner, the width $w_k$ and the aspect ratio $\rho_k = \frac{w_k}{h_k}$ of the rectangular patch (we verified that tracking the aspect ratio instead of the box of the target object in the image plane, as well as its velocity and changes in size). Hence a first order dynamic model, with the following update equation has been used:

$$
\mathbf{x}_k = A\mathbf{x}_{k-1} + \mathbf{v}_{k-1},
$$

where $A$ is an $8 \times 8$ matrix, $I_4$ is the $4 \times 4$ identity matrix, $\Delta t$ is the time step and $\mathbf{v}_{k-1}$ is an additive, zero mean, isotropic Gaussian uncertainty term. In this way, the error $\mathbf{v}_{k-1}$ includes not only the unknown dynamic components (e.g. acceleration), but also those components that we may not want to include or be able to include in the model. Since acceleration is not counted, the capability of the tracker to follow accelerating targets depends on how much uncertainty is modeled with $\mathbf{v}_{k-1}$. This uncertainty can be parameterized in terms of the standard deviation on each component of the state vector:

$$
\Sigma = \begin{bmatrix}
\sigma^x & \sigma^y & \sigma^w & \sigma^\rho & \sigma^\delta^x & \sigma^\delta^y & \sigma^\delta^w & \sigma^\delta^\rho
\end{bmatrix}^T = \begin{bmatrix}
\Sigma^S & \Sigma^D
\end{bmatrix}^T.
$$

where, for the sake of simplicity, we have put into evidence the static ($\Sigma^S$) and dynamic ($\Sigma^D$) part of the vector.

2.2. Measurement equation

The measurement process is based on histogram similarity: the target histogram is compared with that of other candidate patches extracted from the last captured frame, and the most similar one is chosen. Several types of noise can affect the measurement process. First of all, the histogram has some intrinsic weakness due to the fact that no spatial information is taken into account: any permutation of the same set of pixels has the same color histogram. Moreover, the rectangular patch will necessarily include some background pixels, thus adding some noise to the target histogram. Since the target is observed in the image plane (it is just a sampled and quantized projection of the real target) its appearance will also change, in general in non-linear and unpredictable ways. Finally, since the set of candidate patches for the measurement process is finite, the patch that is chosen as the final measure could not be that one that maximizes the histogram similarity over the entire image plane.

According to this, the noise in the measurement process has been modeled as an additive zero mean isotropic Gaussian noise $\mathbf{n}_k$, parameterized with its standard deviation which defines the skewness or smoothness of the measurement PDF. The measurement equation can thus be written as follows:

$$
\mathbf{z}_k = \arg \min_{z \in \mathcal{C}} d(\mathcal{H}(\mathbf{x}_k), \mathcal{H}(z)) + \mathbf{n}_k,
$$

where $\mathcal{C}$ is a set of candidate patches, $\mathcal{H}(\mathbf{x}_k)$ denotes the histogram of the $\mathbf{x}_k$ candidate patch, $\mathcal{H}(\cdot)$ is the target histogram and $d(\cdot, \cdot)$ is a distance in the histograms space. Through the measurement process, at every step the particle filter is provided with a 4-dimensional vector $[\hat{x}_k, \hat{y}_k, \hat{w}_k, \hat{\rho}_k]^T$ that corresponds to the static part of the state vector of the most similar patch in a set of candidate patches, chosen around the last estimated position of the target.

2.3. State estimation

Exploiting the state update Eq. (4) and the measurement Eq. (6), at each iteration $k$ a new set of weighted $N_s$ samples $\{w_i^k, x_i^k\}_{i=1}^{N_s}$ is obtained, each of which represents a localization of the state vector with the tracker confidence. From this set of weighted samples, the estimated value of the state at time $k$ is computed as the weighted mean:

$$
\mathbf{x}_k = \sum_{i=1}^{N_s} w_i^k \mathbf{x}_i^k.
$$

3. Adaptive uncertainty estimation

In the framework introduced in Section 2, there are three different problems to take into account. The first is inherent to the fact that the tracker uses a first order dynamic model. Let us rewrite Eq. (4) in the following form:

$$
\begin{bmatrix} 
\mathbf{s}_k \\
\mathbf{d}_k
\end{bmatrix} =
\begin{bmatrix}
I_4 & I_4 \Delta t \\
0 & I_4
\end{bmatrix}
\begin{bmatrix} 
\mathbf{s}_{k-1} \\
\mathbf{d}_{k-1}
\end{bmatrix} +
\begin{bmatrix} 
\mathbf{v}_{k-1} \\
\mathbf{v}_{k-1}
\end{bmatrix}.
$$

---

*Fig. 1.* Plots of the sigmoid function for different values of $\alpha$ and $\beta$. 
Fig. 2. Test sequences used in our experiments. For each sequence are shown the target trajectory superimposed to the scene image and the plots of velocity angle and module. Velocity angles are displayed in the $[0^\circ,360^\circ]$ interval, so that negative values generate the zig-zag effect in the plots.
Fig. 3. Test sequences used in our experiments. For each sequence are shown the target trajectory superimposed to the scene image and the plots of velocity angle and module. Velocity angles are displayed in the $[0^\circ;360^\circ]$ interval, so that negative values generate the zig-zag effect in the plots.
Table 1

<table>
<thead>
<tr>
<th>Seq</th>
<th>Adaptive Particle Filter tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha = 4$</td>
</tr>
<tr>
<td></td>
<td>$\beta = 0.5$</td>
</tr>
<tr>
<td>($-\sigma$)</td>
<td>($\sigma$)</td>
</tr>
<tr>
<td>1</td>
<td>0.43 0.15</td>
</tr>
<tr>
<td>2</td>
<td>0.35 0.21</td>
</tr>
<tr>
<td>3</td>
<td>0.56 0.13</td>
</tr>
<tr>
<td>4</td>
<td>0.64 0.13</td>
</tr>
<tr>
<td>5</td>
<td>0.39 0.12</td>
</tr>
<tr>
<td>6</td>
<td>0.51 0.14</td>
</tr>
<tr>
<td>7</td>
<td>0.477 0.131</td>
</tr>
<tr>
<td>8</td>
<td>0.539 0.094</td>
</tr>
<tr>
<td>9</td>
<td>0.465 0.149</td>
</tr>
<tr>
<td>10</td>
<td>0.381 0.169</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Seq</th>
<th>Histogram similarity measure</th>
<th>Correlation</th>
<th>Intersection</th>
<th>$\chi^2$</th>
<th>Bhattacharyya</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>($-\sigma$)</td>
<td>($\sigma$)</td>
<td>($-\sigma$)</td>
<td>($\sigma$)</td>
<td>($-\sigma$)</td>
</tr>
<tr>
<td>1</td>
<td>0.30 0.16</td>
<td>0.48 0.20</td>
<td>0.51 0.11</td>
<td>0.44 0.13</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.23 0.23</td>
<td>0.32 0.22</td>
<td>0.32 0.22</td>
<td>0.37 0.20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.26 0.15</td>
<td>0.58 0.12</td>
<td>0.61 0.11</td>
<td>0.57 0.12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.66 0.12</td>
<td>0.78 0.14</td>
<td>0.68 0.13</td>
<td>0.65 0.13</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.20 0.10</td>
<td>0.39 0.11</td>
<td>0.35 0.12</td>
<td>0.40 0.14</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.49 0.09</td>
<td>0.53 0.08</td>
<td>0.58 0.12</td>
<td>0.53 0.15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.077 0.105</td>
<td>0.433 0.153</td>
<td>0.474 0.128</td>
<td>0.496 0.127</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.328 0.096</td>
<td>0.480 0.116</td>
<td>0.483 0.098</td>
<td>0.526 0.096</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.386 0.152</td>
<td>0.530 0.150</td>
<td>0.535 0.156</td>
<td>0.442 0.145</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.086 0.146</td>
<td>0.334 0.183</td>
<td>0.371 0.169</td>
<td>0.344 0.169</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. Plots of tracking mean quality index $\Sigma_k$ of sequence #1 of our test dataset, with 200, 800, and 3200 particles.

Fig. 5. Plots of tracking mean quality index $\Sigma_k$ of sequence #1 of our test dataset with 800 and 3200 particles, and the 95% confidence intervals.

Table 3

<table>
<thead>
<tr>
<th>Seq</th>
<th># of frames</th>
<th>Adaptive Particle Filter</th>
<th>Non-Adaptive Particle Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$sf$</td>
<td>$\Sigma_f$</td>
<td>$\Sigma$</td>
</tr>
<tr>
<td>1</td>
<td>1 96 79</td>
<td>0.48</td>
<td>0.44</td>
</tr>
<tr>
<td>2</td>
<td>2 49</td>
<td>0.46</td>
<td>0.37</td>
</tr>
<tr>
<td>3</td>
<td>3 191</td>
<td>0.58</td>
<td>0.57</td>
</tr>
<tr>
<td>4</td>
<td>4 116</td>
<td>0.65</td>
<td>0.63</td>
</tr>
<tr>
<td>5</td>
<td>5 115</td>
<td>0.46</td>
<td>0.40</td>
</tr>
<tr>
<td>6</td>
<td>6 97</td>
<td>0.56</td>
<td>0.51</td>
</tr>
<tr>
<td>7</td>
<td>7 1283</td>
<td>1.04</td>
<td>0.496</td>
</tr>
<tr>
<td>8</td>
<td>8 937</td>
<td>0.529</td>
<td>0.526</td>
</tr>
<tr>
<td>9</td>
<td>9 933</td>
<td>0.488</td>
<td>0.442</td>
</tr>
<tr>
<td>10</td>
<td>11 1888</td>
<td>0.447</td>
<td>0.344</td>
</tr>
</tbody>
</table>

$s_k = s_{k-1} + d_{k-1} + v_{k-1}$

$s_k = s_{k-1} + d_{k-2} + v_{k-2} + v_{\Sigma_k-1}$.

From the above expression it is clear that the uncertainty $v_{\Sigma_k-2}$ in the dynamic part of the state vector is propagated through to the static part $s_k$, and amplifies the noise on the target position and size. This requires that the relative influence of the standard deviation of the static and dynamic part of the noise must be carefully designed in order to avoid a sensible decrease of accuracy. To be convinced of this let us suppose for the sake of simplicity that the tracked target moves with constant velocity, such that $d_{k-1} \approx d_{k-2}$. In this case, expressions (9) and (10) only differ by the $v_{\Sigma_k-2}$ noise, and equality only holds if $v_{\Sigma_k-2} \approx v_{k-2}$. Should standard deviation $\Sigma_{\Sigma_k}$ of $v_{\Sigma_k-2}$ be similar to $\Sigma_{\Sigma_k-2}$ in the static part $s_k$ of the state vector will be affected by much more noise than expected in the model. Even worse, if $\Sigma_{\Sigma_k}$ is greater than $\Sigma_{\Sigma_k}$, the application of the system update Eq. (4) would upset the estimates of the filter, since the noise on the velocity would dominate that on position. This is avoided if the magnitude of noise on the dynamic part of the state is set high enough to compensate the variations in the target velocity (i.e., acceleration) that are not modeled in the first order dynamic model, yet remaining negligible with respect to the magnitude of noise on the static part. According to this, and based on experimental verification, in our model we have imposed that.
the standard deviation $\Sigma^d$ is at least one order of magnitude less than standard deviation $\Sigma^s$, i.e.:

$$\Sigma^s > \Sigma^d.$$  \hfill (11)

The second problem the particle filter has to deal with is concerned with the fact that there is a relation between the observed target motion and target scale. Since the particle filter spreads its particles in a bounded interval of the state space of size proportional to the noise magnitude (this makes the filter robust to wrong motion estimations), in order to make the tracker capable to track a target moving towards the camera or away from it, it is needed that variations in the target’s size correspond to similar variations in noise magnitude.

Finally, due to the fact that the appearance model is weak (the color histogram), the state PDF may have several modes, corresponding to image patches with appearance model similar to the target though they do not represent the target at all. When this happens, some particles of the filter may gain a high weight, even if they are far from the actual state value, and this can make the estimate very inaccurate, since it is usually computed as the mean value of the distribution. To deal with this problem, it is needed to reduce the size of the state space interval where the filter spreads its particles, whenever it is possible.

A solution to the problems mentioned above is to adapt the standard deviations $\Sigma^s$ and $\Sigma^d$ of the noise $\psi_{k-1}$ in Eq. (1) to the tracker performance. Therefore, in our approach we make these values time-dependent:

$$\Sigma^s_k = [\sigma_k^x \sigma_k^y \sigma_k^w \sigma_k^h]^T, \quad \Sigma^d_k = [\sigma_k^x \sigma_k^y \sigma_k^w \sigma_k^h]^T.$$  

Initial values $\Sigma^s_0$ and $\Sigma^d_0$ are defined considering the specific tracking task, and $\Sigma^s_k$ and $\Sigma^d_k$ are hence computed from $\Sigma^s_0$ and $\Sigma^d_0$ by considering both the scaling of the target (and the consequent changes in the magnitude of apparent motion) and the tracker’s capability of providing accurate estimates of the target dynamics. We decided that $\Sigma^s_k$ and $\Sigma^d_k$ adapt to the estimated target size according to a linear relationship, and adapt to the accuracy of tracking according to the sigmoid function $\zeta(\psi_k):[0; 1] \rightarrow [0; 1]$, defined as:

$$\zeta(\psi_k) = \frac{\text{erf}(x(\psi_k - \beta)) + 1}{2},$$  \hfill (12)

where $\psi_k$ is the difference between the appearance model of the target and the rectangular patch in the estimated position at time step $k$, $x$ and $\beta$ define the steepness and position of the function (see Fig. 1), and erf() is the error function, defined as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt.$$  \hfill (13)

According to this, $\Sigma^s_k$ and $\Sigma^d_k$ are modeled as:

$$\begin{cases} 
\Sigma^s_k = \zeta(\psi_k) \text{min}(w_k, h_k) \Sigma^s_0, \\
\Sigma^d_k = (1 - \zeta(\psi_k)) \text{min}(w_k, h_k) \Sigma^d_0
\end{cases}$$  \hfill (14)

where $\Sigma^s_0 \gg \Sigma^d_0$, $w_k$ and $h_k = w_k - \rho_k$ are respectively the estimated width and height of the target and the quantity $\text{min}(w_k, h_k)$ is used as a measure of the target size. In this way, it can be decided whether or not the tracker should rely on the target estimated

1 Standard deviations can be chosen according to the fact that with Gaussian distribution the probability of drawing a value at $2\sigma$ from the mean value is about 95% (99% for 3$\sigma$).
motion and, depending on the value of \( \zeta \), it is possible to steer the filter to assume a behavior similar to a \textit{random-walk tracker} (searching quasi at random the target to recover from a bad tracking situation taking into little account the estimated dynamics) or a behavior that closely follows the estimated object dynamics. Indeed the behavior of the tracker never reaches the two extreme behaviors, and shows a smooth and continuous adaptation to the quality of the observations. When the tracker behavior assimilates to the \textit{random walk} behavior and magnifies the noise on \( s_k \), it is likely that the particles of the filter are polarized (they are all estimating the same target dynamics, with some slight differences due to the small error modeled on \( d_k \)). In this case, according to Eqs. (11) and (14), \( \Sigma^i \) is negligible with respect to \( \Sigma^e \). However, the magnitude of the estimated motion before the switch could be very high, thus drifting the particles in the last estimated target direction regardless of the actual dynamics. Their polarization can interfere with the process of recovering from a bad tracking situation. A correction to this fact can be provided by artificially weighting the dynamic components of the state vector with the quantity \( (1 - \zeta(\psi_k)) \):

\[
d_k = (1 - \zeta(\psi_k))d_k. \tag{15}
\]

According to this, when the value of \( \zeta \) is high and the uncertainty in position and size is amplified, the dynamics estimated by each particle is reduced to very low values. Instead, when the tracker provides accurate estimates, the value of \( \zeta \) is very close to zero and there is no practical effect. Algorithm 1 summarizes the algorithm for our Adaptive Particle Filter tracker with the solutions described above.

\begin{algorithm}
\begin{algorithmic}[1]
  \Input \( \{x_{k-1}, w_{k-1}, c_{k-1}\}_{i=1}^{N} \)
  \Output \( \{x_i, w_i, c_i\}_{i=1}^{N} \)
  \State \( c_i^0 = 0 \)
  \If{The next two instructions will affect \( \psi \)}
  \State the statistic of \( \psi_{k-1}, \psi_k \)
  \State \( \Sigma^i_k = \frac{(\psi_{k-1})}{\psi_{k-1} - \psi_{k-1}} \Sigma^2 \)
  \EndIf
  \If{from Eq. (14)}
  \State \( \Sigma^i_k = \frac{(1 - \zeta(\psi_k))}{\psi_{k-1} - \psi_{k-1}} \Sigma^2 \)
  \EndIf
  \If{from Eq. (15)}
  \State \( d_k = (1 - \zeta(\psi_k))d_k \)
  \EndIf
  \For{\( i \in [1:N], \)}
  \State \( j = \min \{k \in \{1 \ldots N_i\} | c_{k-1}^j \geq r \} \)
  \State \( \text{resampled particle} \)
  \State \( x_k = \frac{1}{a_k} \sum_{i=1}^{a_k} \tilde{x}_{k-1} \)
  \State \( x_k^i = \tilde{x}_{k-1} \)
  \State \( w_i = p(z_k | x_k) \)
  \State \( c_k^i = c_k^{i-1} + w_i \)
  \State \( \text{compute cumulative sum} \)
  \EndFor
\end{algorithmic}
\end{algorithm}

Fig. 7. Comparative performance analysis: Plots of mean quality index \( Q \) (sequences \#4, \#5 and \#6 of our test dataset). The 95% confidence interval is also plotted.
4. Experiments

The Adaptive Particle Filter tracker described in Section 3 has been implemented in C++ using the OpenCV library. Color histograms are calculated in HSV color space for hue and saturation, in order to have invariance to illumination changes. The original color resolution is downsampled to 3 bits (8 colors) per channel, to reduce the sensitivity to color and improve efficiency. The implementation runs at 25 fps, with 640 × 480 pixels resolution on an AMD Athlon™64 3500 + CPU.

For the experiments reported in this section, we did not use public data such as the Caviar sequences of ECCV 2004 PETS for several reasons. On the one hand, these sequences are recorded with a fish-eye camera that introduces unnecessary strong distortions for the goals of our experiments; on the other hand, many sequences are recorded with people entering/leaving the scene repeatedly, with the goal of verifying the capability of the tracker to manage birth and death processes, which is beyond the scope of this work. Instead, we have built a new set of ten sequences with one moving target, such that it was possible to extract the ground truth reliably.

Test sequences from #1 to #6 are very short sequences (from 2” to 8”) with a single distinguishing motion feature (e.g. left hand turn with almost constant velocity, uniform acceleration, etc.). Test sequences from #7 to #10 have much longer duration (from 30” to 1’) and include different motion conditions with very quick changes, and more critical conditions such as frequent illumination changes, changes of target appearance and partial and total occlusions (sequences #9 and #10).

As a measure of tracking performance at frame \( k \) we have used the quality index proposed by Phillips and Chhabra in [33]:

\[
Q_k = \frac{\sum_{1}^{N_k} \bar{Q}_k}{N_k}
\]

where \( N_k \) is the number of frames in the sequence;

\[
\bar{Q}_k = \frac{|E^g_k \cap E^l_k|}{|E^g_k \cup E^l_k|}
\]

being \( E^g_k \) and \( E^l_k \) are respectively the target bounding rectangle for the ground truth and for the tracker estimation at frame \( k \). Since the particle filter is based on the simulation of a stochastic process, the average \( \bar{Q}_k \) over 20 runs has been used as a meaningful measure of performance at frame \( k \).

For the purpose of our evaluation, we considered two distinct quality indexes:

- The average quality index \( \bar{o} \) for the whole sequence and its standard deviation \( \sigma \) defined as:

\[
\bar{o} = \frac{1}{N_k} \sum_{1}^{N_k} Q_k
\]

being \( N_k = |k| \) the total number of frames in the sequence;

- The average quality index \( \bar{o}_s \) for the set of frames where tracking was successful, defined as the above, but considering only the frames where the target was successfully tracked.

For the experiments reported in this section, we did not use public data such as the Caviar sequences of ECCV 2004 PETS for several reasons. On the one hand, these sequences are recorded with a fish-eye camera that introduces unnecessary strong distortions for the goals of our experiments; on the other hand, many sequences are recorded with people entering/leaving the scene repeatedly, with the goal of verifying the capability of the tracker to manage birth and death processes, which is beyond the scope of this work. Instead, we have built a new set of ten sequences with one moving target, such that it was possible to extract the ground truth reliably.

Test sequences from #1 to #6 are very short sequences (from 2” to 8”) with a single distinguishing motion feature (e.g. left hand turn with almost constant velocity, uniform acceleration, etc.). Test sequences from #7 to #10 have much longer duration (from 30” to 1’) and include different motion conditions with very quick changes, and more critical conditions such as frequent illumination changes, changes of target appearance and partial and total occlusions (sequences #9 and #10).

As a measure of tracking performance at frame \( k \) we have used the quality index proposed by Phillips and Chhabra in [33]:

\[
Q_k = \frac{\sum_{1}^{N_k} \bar{Q}_k}{N_k}
\]

where \( N_k \) is the number of frames in the sequence;

\[
\bar{Q}_k = \frac{|E^g_k \cap E^l_k|}{|E^g_k \cup E^l_k|}
\]

being \( E^g_k \) and \( E^l_k \) are respectively the target bounding rectangle for the ground truth and for the tracker estimation at frame \( k \). Since the particle filter is based on the simulation of a stochastic process, the average \( \bar{Q}_k \) over 20 runs has been used as a meaningful measure of performance at frame \( k \).

For the purpose of our evaluation, we considered two distinct quality indexes:

- The average quality index \( \bar{o} \) for the whole sequence and its standard deviation \( \sigma \) defined as:

\[
\bar{o} = \frac{1}{N_k} \sum_{1}^{N_k} Q_k
\]

being \( N_k = |k| \) the total number of frames in the sequence;

- The average quality index \( \bar{o}_s \) for the set of frames where tracking was successful, defined as the above, but considering only the frames where the target was successfully tracked.
Finally, in order to analyze the capability of the tracker to recover the target in the presence of occlusions, we also considered the average time-to-recover $T_{d}$, computed as the average over the entire sequence of the number of frames elapsed between miss and recovery of the target after each occlusion.

### 4.1. Performance evaluation overview

The influence of $\alpha$ and $\beta$ sigmoid parameters over the performance of our Adaptive Particle Filter tracker is shown in Table 1. It appears that different values of $\alpha$ and $\beta$ parameters have some contrasting influence on the behavior of the tracker. No setting gives superior performance in all the test sequences. For the long sequences that include critical conditions such as illumination changes, cluttered background, non-linear target motion, shape change, partial and total occlusions, with frequent changes of conditions, the tracker will switch often between its two extreme behaviors (following the target, searching for the target). In this case, some performance boost can be obtained by finely tuning the sigmoid parameters. For the experiments in the following we set $\alpha = 8$ and $\beta = 0.5$.

Table 2 reports the average performance of our Adaptive Particle Filter tracker for different histogram similarity measures: correlation, intersection, $\chi^2$ and Bhattacharyya. We can observe that Bhattacharyya and $\chi^2$ similarity measures perform best, consistently with the results in the literature [34]. For the experiments in the following we used the Bhattacharyya distance.

Fig. 4 reports the performance of the tracker as a function of the number of particles $N_{s}$ of the filter. Sequence #1 is used for the sake of example. We observe that there is no significant improvement of performance beyond 800 particles. However, as can be observed in Fig. 5 a higher number of particles reduces the confidence interval and therefore improves the tracker stability. After experiments we have found that $N_{s} = 1000$ tradeoffs computational effort for real-time tracking and tracker performance stability.

The influence of adaptation on the model uncertainty is shown in Table 3, where the performance of the Adaptive Particle Filter tracker is compared with the performance of the same particle filter tracker with no adaptation. The particle filter tracker with no adaptation almost always shows the worst results.

Details of the behaviors of the two trackers can be viewed from the plots of $Q_{k}$ in Figs. 6–11. It can be noticed that uncertainty adaptation gives a very positive contribution especially in the most challenging sequences.

### 4.2. Comparative performance analysis

In the following we compare the performance of our Adaptive Particle Filter tracker with three notable trackers, representative of three different approaches to tracking. Particularly, we compare with the CAM-shift tracker (that applies deterministic tracking), the Zhou’s method [31] (that, similarly to our approach performs detection-by-tracking using particle filtering, but with a different adaptation – on the aspect model and number of particles only) and the Stalder’s method [12] (that performs tracking-by-detection).

A summary of the comparative analysis is reported in Tables 4 and 5. For the computation of $Q_{sf}$, we used different definitions, depending on the different modes of operation of the methods.
compared. In the CAM-shift tracker, the Adaptive Particle Filter tracker by Zhou et al. [31], and our Adaptive Particle Filter tracker, estimation of the target position is provided at every frame. According to this, for these methods successful tracking in a frame is defined by the value of the mean quality index of the frame $Q_k$. A threshold of 0.3 has been considered for all the three methods. Instead, in the Stalder's tracker [12], no hypothesis about the target position is provided by the tracker when the target is not detected by the detector. According to this, with this method tracking is successful in a frame only if the target is detected in that frame.

Plots of $Q_k$ in Figs. 6–11 show the behavior of the trackers in more detail. It can be observed that our Adaptive Particle Filter tracker has slightly worse performance than the CAM-shift tracker for the sequences with no occlusions or little variations (sequences #1, #2, #3, #4 and #7). The reason of this is that the CAM-shift tracker provides much better scale localization (it exploits the full posterior density when seeking for the density global mode) while the Adaptive Particle Filter tracker only exploits a number of samples of the posterior (which could not include the global mode). On the other hand, our Adaptive Particle Filter tracker significantly improves with respect to the CAM-shift tracker in the cases in which target appearance and motion have strong variability in consecutive frames and there are target occlusions (sequences #5, #6, #8, #9 and #10). In fact, in these cases there is high probability that the CAM-shift tracker is trapped in local optima and misses the target. In the presence of occlusions and strong changes of the target appearance (sequences #5 and #10) the Adaptive Particle Filter tracker, although is not as accurate as the CAM-shift tracker in localizing the target, nevertheless shows much more robustness to recover the target. For example, in sequence #10, where occlusions are very frequent, our Adaptive Particle Filter shows a low quality index since there is not enough time to adjust the target size estimation between consecutive occlusions, but nevertheless keeps locked to the target for most of the sequence. We can also observe that the CAM-shift tracker has a non-univocal behavior in the sequences with no occlusions: either it performs slightly better (sequence #7), or awfully worse (sequences #5, #6 and #8) than our Adaptive Particle Filter tracker. In practice, when CAM-shift tracker gets stuck on some local optimum, it is very unlikely that it recovers the target.

Both Zhou’s and Stalder’s methods successfully track the target only in the case of smooth maneuvering and when the target appearance keep almost unchanged from frame to frame. Both show good behavior only for sequence #4, where motion has uniform acceleration with smooth change between consecutive frames. Performance drops are instead observed in all the cases with abrupt changes of motion or target appearance as in the U-turns of sequences #5 and #6 or in fast scale changes as in sequences #1 #2 and #3. They are dramatically unable to adapt to quick changes of motion and appearance of the target as those in the long sequences of the dataset #7, #8, #9 and #10. In these cases our Adaptive Particle Filter tracker outperforms both these methods. However, in the cases in which the target is successfully tracked a better target segmentation is observed with both the Zhou’s adaptive particle filter tracker and the Stalder’s tracking-by-detection approach.

Table 6 shows the values of the average time-to-recover $T_t$ as measured for sequences #9 and #10 for the four methods. The
positive effect of the first order model with uncertainty adaptation on the capability of the tracker to recover from target misses is evident.

The Zhou’s tracker was observed to run at approx 5 fps while the Stalder’s tracker at 8–10 fps, with lower performance when the target is lost. All the other trackers performed at approx 25 fps. We have also checked our Adaptive Particle Filter tracker with the sequences provided by Stalder\(^4\) for a more complete assessment of the context of application of detection-by-tracking based on particle filtering and tracking-by-detection with adaptive detector.

---

**Table 4**
Comparative performance analysis: number of frames with successful tracking (sequences of our test dataset). The best values are evidenced in bold.

<table>
<thead>
<tr>
<th>Seq</th>
<th># of frames</th>
<th>Adaptive Particle Filter</th>
<th>CAM-shift tracker</th>
<th>Zhou’s tracker</th>
<th>Stalder’s tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96</td>
<td>79</td>
<td>90</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>33</td>
<td>43</td>
<td>8</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>191</td>
<td>185</td>
<td>191</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>116</td>
<td>115</td>
<td>114</td>
<td>91</td>
<td>104</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>86</td>
<td>3</td>
<td>38</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>97</td>
<td>89</td>
<td>1</td>
<td>39</td>
<td>53</td>
</tr>
<tr>
<td>7</td>
<td>1283</td>
<td>1204</td>
<td>1272</td>
<td>6</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>937</td>
<td>925</td>
<td>2</td>
<td>87</td>
<td>262</td>
</tr>
<tr>
<td>9</td>
<td>933</td>
<td>794</td>
<td>578</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>1188</td>
<td>761</td>
<td>321</td>
<td>4</td>
<td>52</td>
</tr>
</tbody>
</table>

**Table 5**
Comparative performance analysis: Mean quality index computed on successfully tracked frames $Q_{sf}$ and the whole sequence $Q$ (sequences of our test dataset). The best values are evidenced in bold.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Adaptive Particle Filter</th>
<th>CAM-shift tracker</th>
<th>Zhou’s tracker</th>
<th>Stalder’s tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Q_{sf}$</td>
<td>$Q$</td>
<td>$Q_{sf}$</td>
<td>$Q$</td>
</tr>
<tr>
<td>1</td>
<td>0.48</td>
<td>0.44</td>
<td>0.57</td>
<td>0.55</td>
</tr>
<tr>
<td>2</td>
<td>0.46</td>
<td>0.37</td>
<td>0.70</td>
<td>0.64</td>
</tr>
<tr>
<td>3</td>
<td>0.58</td>
<td>0.57</td>
<td>0.67</td>
<td>0.61</td>
</tr>
<tr>
<td>4</td>
<td>0.65</td>
<td>0.65</td>
<td>0.70</td>
<td>0.69</td>
</tr>
<tr>
<td>5</td>
<td>0.46</td>
<td>0.40</td>
<td>0.54</td>
<td>0.68</td>
</tr>
<tr>
<td>6</td>
<td>0.56</td>
<td>0.53</td>
<td>0.49</td>
<td>0.40</td>
</tr>
<tr>
<td>7</td>
<td>0.513</td>
<td>0.496</td>
<td>0.625</td>
<td>0.622</td>
</tr>
<tr>
<td>8</td>
<td>0.529</td>
<td>0.526</td>
<td>0.390</td>
<td>0.509</td>
</tr>
<tr>
<td>9</td>
<td>0.488</td>
<td>0.442</td>
<td>0.740</td>
<td>0.465</td>
</tr>
<tr>
<td>10</td>
<td>0.447</td>
<td>0.344</td>
<td>0.618</td>
<td>0.598</td>
</tr>
</tbody>
</table>

**Table 6**
Comparative performance analysis: time-to-recover $T_d$ (sequences #9 and #10). The best are evidenced in bold.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Adaptive Particle Filter</th>
<th>Non-Adaptive Particle Filter</th>
<th>CAM-shift tracker</th>
<th>Zhou’s tracker</th>
<th>Stalder’s tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>5.16</td>
<td>920.0</td>
<td>68.20</td>
<td>224.75</td>
<td>39.62</td>
</tr>
<tr>
<td>10</td>
<td>8.26</td>
<td>190.83</td>
<td>53.64</td>
<td>1184.0</td>
<td>57.0</td>
</tr>
</tbody>
</table>

Stalder’s sequences all include targets moving with extremely slow, constant speed motion, and present special (in most cases unrealis-
tic) conditions specifically designed by the authors to demonstrate the capability of their detector to adapt to (smooth) changes of target appearance. Particularly, sequence #1 ("Adaptation") was designed to check adaptation to slowly changing illumination conditions in a real context. Sequence #2 ("BackgroundClutter") was designed to check the capability of the detector to distinguish a moving target from a very cluttered background. Sequence #3 ("DynamicOcclusion") and #5 ("StaticOcclusion") were designed to check the capability of the detector to adapt to the target appearance as it becomes progressively occluded: they respectively show a puppet that is moved until being occluded by an obstacle, and a static bottle that is occluded by a different bottle. Sequence #4 ("Scale"), was designed to check the capability of the detector to adapt to scale changes; here the puppet is moved away and towards the camera, so changing its shape size. The results are reported in Table 7.

With sequence #2 the Adaptive Particle Filter scores a poor performance, mainly due to the fact that a simple and weak appearance model is employed (the target color histogram). Since the target has very small size and the background is very cluttered, a multimodal PDF is obtained in most of the frames and target position and size estimation become very uncertain. The Stalder's detector shows instead a very good capability to extract the target from the background. With sequences #3 and #5, our Adaptive Particle Filter tracker succeeds to follow the target until the occlusion doesn't cover large part of the target (we set a 0.3 matching threshold to assess a frame as successfully tracked by our tracker). The Stalder's detector shows instead a very good capability to extract the target from the background. With sequences #3 and #5, our Adaptive Particle Filter tracker succeeds to follow the target until the occlusion doesn't cover large part of the target (we set a 0.3 matching threshold to assess a frame as successfully tracked by our tracker). The Stalder's detector shows instead a very good capability to extract the target from the background.

4.3. Operation in special conditions

In the following we discuss the behavior of our Adaptive Particle Filter in a few significant and critical operating conditions.

Accelerated motion. Target acceleration is one of the most critical facts for our Adaptive Particle Filter tracker. Since it employs a first order dynamic model, it assumes that the target moves with constant velocity both in direction and magnitude. Changes of the predicted velocity (as in the case of accelerated motion) can therefore only be modeled with the noise \( v_{k} \) in the state update Eq. (4). Due to this, if the apparent acceleration of the target exceeds the noise modeled in the state update equation, the particle filter may fail tracking. Even in the case in which the tracker succeeds to follow the target acceleration, a few frames are needed to correct the motion estimation.

In most of the test sequences (all except sequence #4) the target is already in motion when the tracker is initialized. Due to this, in the first few seconds, the target is faster than estimated by the tracker. Consequently, the tracker performance drops to low values. This is particularly visible in Figs. 6 and 7 in sequences #1, #3, #5 and #6 (the V- and U-shaped intervals at the beginning of the plots). However, it can be observed that the tracker quickly recovers good values after few frames. Fig. 12 shows examples of sequence segments with strong target acceleration (in sequences #2 and #6) and the corresponding behavior of our Adaptive Particle Filter tracker. The target bounding box as estimated at different time instants is shown superimposed on the scene image, with the estimated velocity represented as a proportional segment in the upper-left corner. In Fig. 12a, the target moves very fast towards the camera with a strong acceleration. This results in a fast change of the bounding box size that is hardly followed by the particle filter. In Fig. 12b, the worst target segmentation is provided right after the tracker initialization, when the particle filter provides wrong estimations of the target motion.

Partial and total occlusions. Fig. 13 shows an example of the behavior of our Adaptive Particle Filter tracker in the presence of partial and total occlusions (passing behind one obstacle, and passing behind two consecutive obstacles). Although the estimation of the target bounding box is inaccurate, particularly when the target is partially occluded by the obstacle, it can be noticed that the tracker anyway recovers the target after a few frames. This capability of the tracker is shown in Fig. 14 in comparison with the behavior of the CAM-shift tracker. While the CAM-shift tracker gets stuck on the background when the target gets occluded, the Adaptive Particle Filter tracker keeps on searching at random, recovering the target a few frames shortly after.

PTZ camera sequences. Pan–Tilt–Zoom (PTZ) camera sensors have emerged recently as a powerful device for the observation of wide areas, gaze redirection and zoom control on target details. Due to the camera operations, the field of view shrinks or expands, so that targets change rapidly their scale and aspect, because of the change of camera focal length and/or camera-to-object distance. With these sensors, foreground segmentation by background subtraction cannot be used and tracking needs solutions that also account for quick adaptation to the new conditions of operation. The capabilities of the Adaptive Particle Filter tracker are therefore appropriate for these sequences. Fig. 15 shows the performance of the tracker for a test real world PTZ sequence.5 In this sequence, the target-camera distance varies from approximately 10–40 m. Initially, both the camera and the target are still, followed by a period in which the target slowly moves away and the camera smoothly zooms in onto the target. In this period (frames 0–200), the tracker scores very high values of the quality index. Then the camera performs several abrupt pannings followed by zooming in and out several times (frames 201–450). Camera operations are interpreted as if there were some abrupt accelerations of the target, so resulting into performance drops. Nevertheless the tracker succeeds to recover the target in all the cases. Tracking performance is finally improved when the camera is continuously zooming in onto the target (frames 451–600).

5 Available at http://www.micc.unifi.it/dini/research/particle-filter-based-visual-tracking.

5. Conclusions

In this paper we have proposed an effective solution for particle filter-based tracking with first order state model and adaptation of state model uncertainty. The approach proposed is based on the observation that, while particle filter is extremely appealing for its intrinsic capabilities to adapt to new conditions of operation, adaptation to quick changes of appearance and motion is nevertheless critical. In fact, incorporating dynamics into its state update equation can amplify the noise in the tracker so to decrease its per-

---

**Table 7**

Comparative performance analysis: number of successfully tracked frames (Stalder's test set [12]). The best values are evidenced in bold.

<table>
<thead>
<tr>
<th>Seq</th>
<th># of frames</th>
<th>Stalder's tracker</th>
<th>Adaptive Particle Filter tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>396</td>
<td>396</td>
<td>396</td>
</tr>
<tr>
<td>2</td>
<td>1528</td>
<td>760</td>
<td>169</td>
</tr>
<tr>
<td>3</td>
<td>1709</td>
<td>1251</td>
<td>734</td>
</tr>
<tr>
<td>4</td>
<td>1911</td>
<td>616</td>
<td>1342</td>
</tr>
<tr>
<td>5</td>
<td>1735</td>
<td>929</td>
<td>356</td>
</tr>
</tbody>
</table>

---

A. Del Bimbo, F. Dini / Computer Vision and Image Understanding 115 (2011) 771–786
formance to unacceptable levels: dynamics imparts a momentum to the filter estimates, so that as the filter misses the target the momentum hastens to track in the wrong direction. Incorporating target dynamic into the state model with adaptation of the state uncertainty allows the tracker to switch between two different behaviors, either following the estimated dynamics (according to

Fig. 12. Example of accelerated motion: target bounding box estimated at sampled time instants of sequence #2 (a) and sequence #6 (b) of our test dataset. Time instants are marked in the $Q_k$ plots; "0" marks indicate tracker initialization.

Fig. 13. Examples of tracking with occlusions: target bounding box estimated at different time instants of sequence #9 (a) and sequence #10 (b) of our test dataset. Time instants are marked in the $Q_k$ plots; "0" marks indicate tracker initialization.
a first order dynamic model) or acting as a random-walk tracker. The experimental evaluation presented shows that this behavior is particularly effective in video sequences where appearance and motion have quick changes, and there are partial or full target occlusions.

Acknowledgments

We thank Dr. Andrew D. Bagdanov and Dr. Walter Nunziati, former postdocs at University of Firenze, for their contributions to this research. We also acknowledge Prof. R. Chellappa and Dr. Ming Du for their kind support and help in providing a working implementation of the Zhou’s tracker [31].

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cviu.2011.01.004.

Fig. 14. Examples of tracking with occlusions: different behaviors of the Adaptive Particle Filter tracker (top row images) and the CAM-shift tracker (bottom row images) (frames from sequence #10 of our test dataset).

Fig. 15. Adaptive Particle Filter tracking with a sample PTZ camera sequence: Mean quality index $Q_k$ with the 95% confidence interval (top row) is displayed with a few screenshots of the sequence (bottom row).

References

[8] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, Steve Shafer, Multi-camera multi-person tracking for easyliving, in: VS ’00:


S. Stalder, H. Grabner, and L. Van Gool, Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. OLCV 09: 3rd On-line learning for Computer Vision Workshop, Kyoto, Japan, September 2009.


Z. Chen, Bayesian Filtering: From Kalman Filters to Particle Filters and Beyond, Technical Report, Adaptive Systems Lab., McMaster University, Hamilton, ON, Canada, 2003.


