Adaptive uncertainty estimation for particle filter-based trackers

Andrew D. Bagdanov, Alberto Del Bimbo, Fabrizio Dini, Walter Nunziati
{bagdanov, delbimbo, dini, nunziati}@dsi.unifi.it
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze
Via di Santa Marta 3, 50139 Firenze, Italy

Abstract

In particle filter–based visual trackers, dynamic velocity components are typically incorporated into the state update equations. In these cases, there is a risk that the uncertainty in the model update stage can become amplified in unexpected and undesirable ways, leading to erroneous behavior of the tracker. To deal with this problem, we propose a continuously adaptive approach to estimating uncertainty in the particle filter, one that balances the uncertainty in its static and dynamic elements. We provide quantitative performance evaluation of the resulting particle filter tracker on a set of ten video sequences. Results are reported in terms of a metric that can be used to objectively evaluate the performance of visual trackers. This metric is used to compare our modified particle filter tracker and the continuously adaptive mean shift tracker. Results show that the performance of the particle filter is significantly improved through adaptive parameter estimation, particularly in cases of partial and total occlusions and erratic, non-linear target motion.

1. Introduction

Tracking of visual phenomena in video sequences is an important task in many computer vision applications. Visual surveillance, video retrieval, human-computer interaction, and many other computer vision applications require robust tracking of moving objects in order to function optimally. In order to be generally applicable in vision systems, visual trackers must be robust to total and partial occlusion, target scale variation, highly erratic target motion, and noisy measurements. Furthermore, for many applications such as video surveillance, trackers must provide reliable tracks over very long sequences. Finally, in active camera and human-computer interaction applications trackers must additionally be very efficient and operate in real-time at very high frame rates.

A tracking technique that has received considerable attention is the family of particle filter-based trackers[2, 4, 5]. The particle filter is a technique based on recursive Bayesian estimation of an unknown system state. Particle filters have attracted so much interest because of their ability to track multiple hypotheses, because they admit simple and efficient implementations, and because they do not require background subtraction in order to function properly.

Despite their popularity, the particle filter approach has several disadvantages. First, many parameters must be specified in the particle filter, and in many cases it is unclear how these should be chosen for a particular problem. Often, no single set of parameters is appropriate for tracking over long video sequences. Furthermore, the lack of a systematic, large-scale empirical study of the performance of particle filter–based visual trackers makes it difficult to assess their true potential for a visual tracking task.

In this paper we describe a particle filter–based tracker that adapts some of its parameters online in order to improve tracking performance over long sequences. The technique balances a trade off between the uncertainty parameters on the static and dynamic components of the state-space representation of the visual tracking problem. A general and efficient histogram–based technique is adopted for target description. We illustrate the performance of the tracker on a set of calibrated video sequences from which ground truth tracking data can be easily extracted. Results show how the proposed approach improves on the basic particle filter tracker, while preserving the efficiency of the method despite the additional machinery. A comparative evaluation between our particle filter tracker and the continuously adaptive mean shift tracker [3, 1] is also provided.

In the next section we introduce the theory of particle filter tracking and the specific model we use to apply the particle filter to track visual phenomena. In section 3 we describe the technique used to adaptively estimate the noise parameters in the particle filter algorithm. Section 4 contains an extensive discussion of the experiments performed to assess the advantages and disadvantages of the particle filter.
2. Visual tracking with the particle filter

In this section we describe the particle filter and how it can be used to track visual phenomena in video sequences.

2.1. Recursive Bayesian filtering

The particle filter is an estimation technique in the recursive Bayesian estimator family of filters. As with all Bayesian filter techniques, it is based on a system of time-dependent model and measurement equations:

\[ x_k = f_k(x_{k-1}, v_{k-1}) \]
\[ z_k = h_k(x_k, n_k). \]

where \( x_k \) denotes the state of the system at time \( k \), and \( z_k \) denotes the measurement of the unknown system state at time \( k \). Note that here we implicitly assumed the system is Markovian, i.e. that \( f_k \) depends only on the previous time step \( k-1 \).

Equation (1) is the system update equation, and represents the evolution of the state of the system from time \( k-1 \) to time \( k \). It depends on the previous states of the system, denoted by \( x_{k-1} \) and a stochastic error \( v_{k-1} \) that represents the uncertainty in the state update. Equation (2) represents a measurement of the unknown state \( x_k \) at time \( k \). It is parametrized in terms of the current (unknown) state \( x_k \) and an error term \( n_k \) representing the uncertainty in measuring the state.

The state space for our tracker is defined over vectors of the form:

\[ \mathbf{x} = \begin{bmatrix} x & y & w & \rho & \dot{x} & \dot{y} & \dot{w} & \dot{\rho} \end{bmatrix}^T. \]

The state vectors are naturally split into two components. The static part, \( s = [x \ y \ w \ \rho] \), specifies the position and size of the tracked object. The dynamic component, \( d = [\dot{x} \ \dot{y} \ \dot{w} \ \dot{\rho}] \), specifies the velocities of the static elements in \( s \).

We track rectangular patches in the images which are specified by their upper left corner \((x, y)\) and their width and aspect ratio \((w, \rho)\). We discovered that tracking aspect ratio instead of width and height independently resulted in fewer degenerate cases. The system update equation we use is thus:

\[ x_k = (s_{k-1} + d_{k-1}) + v_{k-1}, \]

where \( s_{k-1} \) and \( d_{k-1} \) represent the static and dynamic components of the state at time \( k-1 \), respectively, and \( v_{k-1} \) is an additive, zero mean, isotropic Gaussian uncertainty term. This uncertainty is parametrized in terms of the standard deviation on each component of the state vector:

\[ (\sigma_x, \sigma_y, \sigma_w, \sigma_{\rho}, \sigma_{\dot{x}}, \sigma_{\dot{y}}, \sigma_{\dot{w}}, \sigma_{\dot{\rho}}) \]

In our tracker, we use histogram similarity as a measurement model. An estimate, which is a rectangle in the image plane, is used to extract the local histogram corresponding to the estimate. This is compared to the initial target histogram with which the tracker was initialized. The Bhattacharyya similarity measure between probability densities is commonly used for this.

\[ d(H_1, H_1) = \sum_i \sqrt{H_1(i) H_1(i)}. \]

We pass this value through a Gaussian in order to smooth it and add uncertainty.

The Bayesian estimator for the unknown state \( x_k \) at time \( k \) is derived from the state update equation (1), the measurement equation (2), and the known (assumed) statistics of the noise parameters \( v_{k-1} \) and \( n_k \):

\[ p(x_k | z_k) = \frac{p(z_k | x_k)p(x_k | z_{k-1})}{p(z_k | z_{k-1})}, \]

where \( p(x_k | z_{k-1}) \) is derived from the Markov assumption and the Chapman-Kolmogorov equation:

\[ p(x_k | z_{k-1}) = \int p(x_k | z_{k-1})p(x_{k-1} | z_{k-1})dx_{k-1}. \]

Only under very restrictive assumptions about the form and statistics of (1) and (2) can an analytic form of the integral in (7) be found. The particle filter is a stochastic, Monte Carlo approach to solving this problem.

2.2. The particle filter

The particle filter approach to the estimation problem is to use a set of points from the state space \( \{x_k^i|N_k\} \) and a set of accompanying weights \( \{w_k^i|N_k\} \) to form a weighted estimate of the posterior:

\[ p(x_k | z_k) \approx \sum_{i=1}^{N_k} w_k^i \delta(x_k - x_k^i), \]

The theory of importance sampling ensures that we can construct such an estimator as long as each sample \( x_k^i \) (known as a particle), is chosen and weighted according to:

\[ x_k^i \sim q(x_k^i | x_k^{i-1}, z_k) \]
\[ w_k^i \propto w_{k-1}^i p(z_k | x_k^i)p(x_k^i | x_k^{i-1}) \]

The distribution \( q(x_k | x_k^{i-1}, z_k) \) is known as the proposal distribution and in our case is chosen to be equal to be equal to the prior \( p(x_k | x_k^{i-1}) \). Substituting this into (9) gives a weight of:

\[ w_k^i \propto w_{k-1}^i p(z_k | x_k^i). \]
The importance density (the prior) is determined by the system update equation (3) and the known uncertainty statistics (4). The measurement density \( p(z_k | x_k^i) \) is given by the similarity of the histogram of the image patch indicated by the particle \( x_k^i \) and the original target histogram.

### Algorithm 1: PARTICLE FILTER TRACKER

**Input:** \( \{ x_{k-1}^i, w_{k-1}^i, c_{k-1}^i \}_{i=1}^{N_s}, z_k \)

**Output:** \( \{ x_k^i, w_k^i, c_k^i \}_{i=1}^{N_s} \)

1. \( c_k^0 = 0; \)
2. for \( i \in [1; N_s] \) do
   3. \( j = \min \{ l \in \{1 \ldots N_s \} | c_{l-1}^i \geq r \}; \)
   4. \( \tilde{x}_{k-1}^i := \tilde{x}_j^i; \)
   5. \( x_k^i := f_k(\tilde{x}_{k-1}^i, v_{k-1}); \) \hspace{1cm} (from equation 3)
   6. \( w_k^i = p(z_k | x_k^i); \) \hspace{1cm} (from equation 5)
   7. \( c_k^i = c_{l-1}^i + w_k^i; \)
3. endfor
4. for \( i \in [1; N_s] \) do
   5. \( w_k^i = \frac{w_k^i}{c_k^i}; \)
   6. \( c_k^i = \frac{c_k^i}{c_k^i}; \)
5. endfor

Pseudocode for the a single iteration of the particle filter tracker is given in algorithm 1. Lines 3-5 and 8 represent a phase of resampling in which the particles are resampled according to their weight. This prevents impoverishment of the particle set, in which only a few particles of non-negligible weight remain. Lines 10-12 re-normalize the weights so they sum to one.

### 3. Adaptive uncertainty estimation

Trackers are often developed and evaluated under particularly restrictive conditions. Typically they are evaluated on a small number of relatively short sequences. In such situations, a single set of parameters can usually be found with which the tracker performs satisfactorily on the sequences. For some of these parameters there may not be a reasonable way of estimating them online, and there may not even be any clear indication of how they should be selected a priori for a specific tracking task. This limits the general application of such trackers, and further limits their application to longer sequences containing a wide variety of non-linear phenomena, illumination changes, etc.

For particle filter trackers, the most important parameters controlling performance are the noise variances in equation (4). Consider the case where the static variances are set to very high values. In this case increasing the noise on \((\sigma_x, \sigma_y, \sigma_w)\) ensures that the filter samples over a wide enough area to maximize the possibility of capturing the target in case of erratic changes in direction or velocity. The pitfall in this strategy, however, is that it also increases the likelihood that the particle filter will become distracted by spuriously similar patches in the background.

Consider also the uncertainty imposed on the dynamic part of the state vector \((\sigma_{x\cdot}, \sigma_{y\cdot}, \sigma_{w\cdot})\). From equation (3), the update equation for propagating a particle from time \( k - 1 \) to \( k \) is, in the \( x \) component only and after substituting the expression for \( \tilde{x}_{k-1} \):

\[
x_k = x_{k-1} + \tilde{x}_{k-1} + v_x, k - 2 + v_{x, k - 2}.
\]

In this way we see that the uncertainty in the dynamic component is propagated through to the static component, amplifying the noise on the position and size. Even worse, this uncertainty estimate is based on information at iteration \( k - 2 \) and not only on the previous iteration. In any case, the noise on the static and dynamic elements should not both be set to elevated values.

To control the tradeoff between static and dynamic uncertainty, we use the similarity of the current estimate with the original target histogram. Letting \( s_k \) denote the current similarity, we obtain a blindness value by passing it through a sigmoid:

\[
\zeta(s_k) = \frac{\text{erf}(\alpha((1 - s_k) - \beta)) + 1}{2}
\]

The parameter \( \alpha \) controls the steepness of the transition, and \( \beta \) adjusts the position at which the transition takes place.

At each iteration of the particle filter, we adjust the noise variances according to the current blindness estimated at the previous iteration:

\[
\begin{bmatrix}
\sigma_x^k \\
\sigma_y^k \\
\sigma_w^k
\end{bmatrix} =
\begin{bmatrix}
\zeta & 0 & 0 & 0 & 0 \\
0 & \zeta & 0 & 0 & 0 \\
0 & 0 & \zeta & 0 & 0 \\
0 & 0 & 0 & 1 - \zeta & 0 \\
0 & 0 & 0 & 0 & 1 - \zeta
\end{bmatrix}
\begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_w
\end{bmatrix}
\]

Here \( \zeta \) denotes the blindness at the previous iteration \( \zeta(s_k) \). This adaptive technique adjusts the variances in such a way that the noise in the static state observations is never amplified due to noise in the dynamic components.

### 4. Experiments

We performed a number of experiments on test sequences recorded in our laboratory in order to asses and study the performance of the standard particle filter tracker and the adaptive version we proposed above. The test sequences were deliberately designed to allow background segmentation to be used in order to extract ground truth
tracking data. The sequences used in our experiments are of a non-trivial length, and contain phenomena classically difficult for trackers such as changing illumination over the target trajectory, highly nonlinear target motion, partial and total occlusions, and cluttered backgrounds. Experiments were designed to analyze the performance of the adaptive particle filter tracker in terms of its parameters, and to provide an evaluation of its performance in comparison to a mean shift tracker.

The tracker described above was implemented in C++ using the image processing OpenCV library. The implementation is extremely efficient and runs in real-time at a frame rate of 25fps on an AMD Athlon(tm) 64 Processor 3500+. Unless stated otherwise, N$_s = 1000$, $\alpha = 8$, $\beta = 0.5$, and the Bhattacharyya histogram similarity metric were used all runs of the tracker. The uncertainty terms in the state update were adapted as described in section 3. Histograms were computed in the hue and saturation planes in order to provide robustness to illumination changes.

### 4.1. Test sequences and ground truth

All sequences used in our performance evaluation are of a remote-controlled car moving on the floor of our laboratory. Videos were shot from two camera angles: overhead and wide-angle. Table 4.1 gives an overview of the ten sequences used in the experiments reported here. The videos were taken using an Axis 207 IP camera, which provides video at 25fps at a resolution of $640 \times 480$. The sequences are divided into two groups. The first six sequences in table 4.1 are relatively short and simple sequences. These were used to calibrate the tracker to estimate the base parameters used in both the particle filter and mean shift tracker. The second group of videos are much longer and contain examples of occlusion and highly erratic and nonlinear target motion.

All of the test sequences were designed such that background subtraction can be used to extract ground truth tracking data for use in comparative performance evaluation. Background subtraction is used to segment the car from each frame, and the bounding rectangle is used as the ground truth for the frame.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47(s)</td>
<td>Circular trajectory, change of direction</td>
</tr>
<tr>
<td>2</td>
<td>78(s)</td>
<td>Car approaching</td>
</tr>
<tr>
<td>3</td>
<td>37.5(s)</td>
<td>Partial trajectory, partial occlusion</td>
</tr>
<tr>
<td>4</td>
<td>53(s)</td>
<td>Vertical trajectory, reversal</td>
</tr>
<tr>
<td>5</td>
<td>52(s)</td>
<td>Partial and total occlusions</td>
</tr>
<tr>
<td>6</td>
<td>47.5(s)</td>
<td>Circular trajectory, partial occlusion</td>
</tr>
<tr>
<td>7</td>
<td>52(s)</td>
<td>Partial and total occlusions</td>
</tr>
<tr>
<td>8</td>
<td>52(s)</td>
<td>Partial trajectory, partial occlusion</td>
</tr>
<tr>
<td>9</td>
<td>52(s)</td>
<td>Circular trajectory, partial occlusion</td>
</tr>
<tr>
<td>10</td>
<td>52(s)</td>
<td>Partial and total occlusions</td>
</tr>
</tbody>
</table>

### Figure 1. Performance and stability in number of particles.

For performance evaluation we adopt the measure proposed by Phillips and Chhabra for evaluation of graphics recognition systems [6]. Letting $E_g^k$ and $E_t^k$ denote the estimated rectangle of the ground truth and the tracker at frame $k$, respectively, the performance measure is given by:

$$Q(E_g^k, E_t^k) = \frac{|E_g^k \cap E_t^k|}{|E_g^k \cup E_t^k|}$$  \hspace{1cm} (11)

This measure is zero only at frames where the tracker and ground truth rectangles are completely disjoint, and is one only when they exactly coincide. An average performance measure, $\overline{Q}$, is obtained by averaging $Q(E_g^k, E_t^k)$ over all the frames in a video sequence.

### 4.2. Results

We first performed an experiment to analyze the sensitivity of the particle filter tracker to the number of particles $N_s$. Figure 4.2 illustrates the performance of the tracker over a range of particle numbers on sequence #1. Due to the stochastic nature of the particle filter, all performance results reported in this paper are averaged over twenty runs of the particle filter tracker on the test sequence. We observe from this that there is not a great difference in tracker performance once there are 400 particles or more. Also shown, however, is the performance of the particle filter tracker with 3200 and 200 particles along with the corresponding, 95% confidence intervals estimated from the twenty runs. We see here that though the performance itself is not greatly affected, the stability of the tracker is improved with additional particles.

In table 2 are reported the average performance statistics, $\overline{Q}$, for the adaptive particle filter tracker and the standard, non-adaptive version of the particle filter tracker. This table is best interpreted in conjunction with the plots in figure 4.2, which show the frame-by-frame performance of the trackers on each sequence. Also shown in the table and plots is the performance of the mean shift tracker on the same sequences. The first thing to note in these results is that uncertainty adaptation significantly increases the performance.
of the particle filter tracker. A slight performance boost can be obtained by tuning the sigmoid parameters, particularly in long sequences where the tracker must switch often between the two behaviors.

Table 4.2 reports the average performance of the tracker for four commonly use histogram similarity measures. The performance of a mean shift tracker on the same sequences is also shown in table 4.2. The mean shift tracker was also run on the hue and saturation planes, using the same histogram bin quantization as the particle filter. From this table we can conclude that Bhattacharyya and $\chi^2$ similarity measures perform best, which is consistent with results found in the literature.

In many cases, particularly on the shorter sequences, the mean shift tracker outperforms the adaptive particle filter. The reason for this is that the mean shift tracker, in general, provides much better scale localization than the particle filter. The reason why the particle filter performance is so low (but not zero) for sequence #2 in figure 4.2 is that the particle filter converges to a very small patch on the target. The mean shift tracker is able to provide better scale localization because it can act decisively and jump to the local mode in the posterior. In many sequences, however, that the mean shift tracker erroneously converges to a local maximum and loses the target. An example of this is shown in figure 4.2, where we see that the adaptive particle filter is able to recover the target after occlusion, while the mean shift tracker gets lost in the background.

5. Discussion

In this paper we have proposed an adaptive technique for estimating the uncertainty parameters in particle filter–based trackers. The approach is based on the observation that particle filters incorporate dynamics into their state update equations run the risk of amplifying noise in the tracker to an unacceptable level. By using the histogram similarity between the current tracker target estimate and the original target, we are able to smoothly balance the tradeoff between uncertainty in velocity, position and size.

We proposed two performance metrics that can be used to objectively evaluate the performance of trackers. They provide a detailed picture of tracker performance at every point in a sequence as well as an average performance value over a whole video. Quantitative experimental evaluation shows that there are benefits to such a tradeoff, particularly in long video sequences and in sequences with occlusions. The comparative evaluation of our tracker and the mean shift tracker shows that the adaptive particle filter copes with occlusion and background clutter more effectively, again particularly in long sequences, than the mean shift tracker.

Observing the results of the particle filter and mean shift trackers, it appears that the situations where they fail are to a large degree disjoint. This indicates that there is potential in approaches incorporating aspects of mode–seeking in, and stochastic approximation of the posterior of the Bayesian estimation problem. Future work will include an investigation of this possibility.

References


Figure 2. Performance of particle filter and mean shift on short sequences

(a) frame 939  (b) frame 951  (c) frame 969

Figure 3. An example sequence with occlusions.