Using Visual Saliency for Object Tracking with Particle Filters
Désiré Sidibé, David Fofi, Fabrice Mériaudeau

To cite this version:
Désiré Sidibé, David Fofi, Fabrice Mériaudeau. Using Visual Saliency for Object Tracking with Particle Filters. EUSIPCO 2010. 18th European Signal Processing Conference, Jul 2010, Aalborg, Denmark. pp.1776-1780. hal-00584682

HAL Id: hal-00584682
https://u-bourgogne.hal.science/hal-00584682
Submitted on 11 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
ABSTRACT
This paper presents a robust tracking method based on the integration of visual saliency information into the particle filter framework. While particle filter has been successfully used for tracking non-rigid objects, it shows poor performances in the presence of large illumination variation, occlusions and when the target object and background have similar color distributions. We show that considering saliency information significantly improves the performance of particle filter based tracking. In particular, the proposed method is robust against occlusion and large illumination variation while requiring a reduced number of particles. Experimental results demonstrate the efficiency and effectiveness of our approach.

1. INTRODUCTION

Tracking moving objects is an important task in many computer vision applications including video surveillance, smart rooms, mobile robotics, augmented reality and video compression. Despite many effort, it is still a challenging problem due to the presence of noise, changes of illumination, cluttered background and occlusions that introduce uncertainty in the estimation of the object’s state.

The main objective of tracking is to roughly predict and estimate the location of a target object in each frame of a sequence. Many methods have been developed and can be divided into two groups: deterministic methods and stochastic methods [12]. Methods of the former group iteratively search for the local maxima of a similarity measure between the template of the target and the current image. The Kanade-Lucas-Tomasi tracker [7] and the mean-shift tracker [3] are examples of this category of methods. In contrast, methods of the latter group use the state space representation of the moving object to model its underlying dynamics. The tracking problem can then be viewed as a Bayesian inference problem. In the case of a linear dynamic model with Gaussian noise, Kalman filter provides an optimal solution. However, for non-linear and non-Gaussian cases, it is impossible to find analytic solutions. Over the last decade, particle filters, also known as condensation or sequential Monte Carlo methods, have proved to be very efficient for object tracking. For more details and theoretical proofs, the reader is invited to refer to [4, 2]. A particle filter is a sequential Monte Carlo method, which recursively approximates the posterior distribution using a finite set of weighted samples \( \{x_i, w_i\}_{i=1,...,N} \). Each sample \( x_i \) represents a hypothetical state of the target with a corresponding importance weight \( w_i \). Given all observations up to time \( t \), \( Z_t = \{z_0, z_1, ..., z_t\} \), the goal is to estimate the state of the target object \( x_t \), i.e. to find the posterior distribution \( p(x_t | Z_t) \). Let’s assume that the system is governed by the following state-space representation:

\[
\begin{align*}
    x_t &= f(x_{t-1}) + v_{t-1} \\
    z_t &= h(x_t) + n_t
\end{align*}
\]

where \( f \) and \( h \) are respectively the system transition and the measurement functions, and \( v_{t-1} \) and \( n_t \) are the system and measurement noises.

The particle filter, like any sequential Bayesian technique, uses a prediction and correction strategy. The pre-
diction stage uses the system transition model to predict the posterior at time \( t \) as:

\[
p(x_t | Z_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | Z_{t-1}) dx_{t-1}.
\]

(2)

The correction step uses the available observation at time \( t, z_t \), to update the posterior using Bayes’ rule:

\[
p(x_t | Z_t) = \frac{p(z_t | x_t) p(x_t | Z_{t-1})}{p(z_t | Z_{t-1})}.
\]

(3)

### 2.1 Color Distribution Model

Different types of features can be used to measure the observation likelihood of the samples. Among them, color distribution is robust against noise and partial occlusion, and fast to compute [9, 8]. Usually, color distributions are represented by histograms in the RGB or HSV color space. The color distribution \( p(x) = \{p_u(x)\}_{u=1,...,m} \) of a region centered at location \( x \) is given by:

\[
p_u(x) = C \sum_{i=1}^{N_u} k(||x_i - x||^2) \delta[b(x_i) - u],
\]

(4)

where \( C \) is a normalizer, \( \delta \) is the Kronecker function, \( k \) is a kernel with bandwidth \( h \), \( N_u \) is the number of pixels in the region and \( b(x_i) \) is a function that assigns one of the \( m \)-bins to a given color at location \( x_i \). The kernel \( k \) is used to consider spatial information by lowering the contribution of farther pixels.

Several distances can be defined to compute the similarity between color distributions such as the KL-distance or histogram intersection. Here, we adopt the popular Bhattacharyya coefficient as a similarity measure [3]. If we denote \( p^* = \{p^*_u(x_0)\}_{u=1,...,m} \) as the reference color model of the object and \( p(x) \) as a candidate color model, then the distance between \( p^* \) and \( p(x) \) is defined by:

\[
\rho[p^*, p(x)] = \left[ 1 - \sum_{u=1}^{m} \sqrt{p^*_u(x_0) p_u(x)} \right]^{\frac{1}{2}}
\]

(5)

Each sample \( x_i \) is assigned an importance weight which corresponds to the likelihood that \( x_i \) is the true location of the object. In the case of the bootstrap filter [4], the weights are given by the observation likelihood:

\[
w^*_i = p(z_t | x_i^*) \propto e^{-\lambda \rho[p^*, p(x_i^*)]^2},
\]

(6)

where \( \lambda = 20 \) in our experiments as in [9].

### 3. VISUAL SALIENCY BASED TRACKING

In particle filtering based tracking, one has to resolve the contradiction between robustness and tracking speed. In fact, a large number of particles, or samples, leads to more robust results at the price of high computational load and slow tracking speed. Moreover, the features distributions have to be evaluated for each of the samples. Based on this considerations, we use the saliency detection method of Achanta et al. [1] in our work. This method is computationally efficient while providing full resolution saliency maps.

3.1 Saliency Detection

The saliency detection method fully described in [1] is based on color and luminance features. For each pixel of the image, we compute the degree of saliency with respect to its neighborhood as the Euclidean distance between the pixel vector and the average vector of the input image in the Lab color space. Formally, the input image \( I \) is first converted to CIE Lab color space \( I' \). From \( I' \), one computes the mean image feature \( I'_\mu = [L_\mu, a_\mu, b_\mu]^T \) and a Gaussian blurred image \( I'_\sigma \) using a \( 5 \times 5 \) separable binomial kernel. The saliency at a pixel location \((x, y)\) is then given by:

\[
S(x, y) = \| I'_\mu - I'_\sigma(x, y) \|,
\]

(7)

where \( \| \cdot \| \) is the \( L_2 \) norm.

The method emphasizes the largest salient objects and generates sharper and well-defined boundaries of salient objects.

Some saliency detection results are shown in Figure 1. As it can be seen, regions that stand out relative to their neighbors are detected as been salient parts of the image. However, in some situations, the object of interest might be detected as less salient. This is depicted in Figure 1-c, where pixels belonging to the face of the person have lower saliency values compared to background pixels. Therefore, saliency information has to be carefully combined with color information to achieve good tracking results. The next subsection explains how we combine these two information.

3.2 Using Saliency for Tracking

Based on the saliency detection method described in Section 3.1, we define a saliency distribution for a region of the image in a similar way to the color distribution, i.e. using equation (4). The similarity between two saliency distributions is measured by the Bhattacharyya distance.

Figure 2 shows examples of similarity measures between color and saliency distributions. As we can notice, in cases where the object of interest and the background have similar colors, color feature is not enough to identify the object. In the given example, the distance between the color distributions of the reference model in Figure 2-a and the
candidate model in Figure 2-b is $\rho_{12} = 0.4983$ while the distance between the color distributions of the reference model and the candidate model in Figure 2-c is $\rho_{13} = 0.4933$. It is thus, hard to distinguish the correct location of the object. Using saliency distributions, the distances are respectively, $\rho_{12} = 0.1963$ and $\rho_{13} = 0.3620$, showing the distinctiveness of visual saliency.

Despite the distinctiveness of saliency feature, in some situations, the object of interest might be detected as being less salient than the background, as mentioned in Section 3.1. Therefore, in order to improve the robustness of the tracker, we combine both color and saliency features, automatically weighting their respective contribution to the likelihood function.

More precisely, given $N$ samples $\{x'_i\}_{i=1,...,N}$ at time $t$, let $\rho'_i$ be the distance between the reference and the $i$-th candidate color distributions, and let $\rho''_i$ be the distance between the reference and the $i$-th candidate saliency distributions. Then each sample $x'_i$ is assigned an importance weight given by:

$$w_i \propto \left(1 - \alpha_i \right)e^{-\lambda \left(\rho''_i\right)^2} + \alpha_i e^{-\lambda \left(\rho'_i\right)^2}. \tag{8}$$

The weighting parameter $\alpha_i$ is evaluated at every time $t$ using the following formula:

$$\alpha_t = \frac{\bar{\rho}_C}{\bar{\rho}_C + \bar{\rho}_S}, \tag{9}$$

where $\bar{\rho}_C$ is the mean value of $\{\rho'_i\}_{i=1,...,N}$.

By employing a time varying weighting parameter, the tracker can adaptively give importance to one feature or the other based on the color and saliency distributions of every frame of the sequence. Thus, we can deal with large illumination variations and similar background color.

### 3.3 Particle Filter Tracking

To implement the particle filter, one has to define the state vector and the dynamic model of the system. We define the state as $x = [x, y, s_x, s_y]^T$, where $(x, y)$ is the location of the target object, $s_x$ and $s_y$ are the scales in the $x$ and $y$ directions. In the prediction stage of the particle filter, the samples are propagated through a dynamic model. We use a first order auto-regressive (AR) process for simplicity:

$$x_t = Ax_{t-1} + v_{t-1}, \tag{10}$$

where $v_{t-1}$ is a multivariate Gaussian random noise and $A$ defines the deterministic system model. A constant velocity model is usually used for the dynamic model.

In the update stage, the observation likelihood for each sample, i.e., the weights for each sample, are estimated using equation (8). In practice, to avoid degeneracy, i.e. all but one particle having negligible weights after a certain number of recursive steps, a bootstrap resampling is performed. The resampling step is also designed to handle sample impoverishment, i.e. particles that have high weights are statistically selected more often than others [2].

### 4. EXPERIMENTS

To evaluate the performance of the proposed tracking method, we applied it to different sequences showing confusing background color and large illumination variations. The image size of the sequences is $320 \times 240$. We use the HSV color space with a $8 \times 8 \times 8$ bins histogram to represent the color distribution, and we employ a 16 bins histogram for saliency distribution. The HSV color space is used instead of RGB because it is less sensitive to lighting conditions. In all experiments, the tracked object is manually initialized in the first frame.

To show the robustness of our method against similar background color, we use the Plane sequence of 300 frames and compare the basic color based particle filter with the saliency based one. In both cases, we use 100 particles for tracking. The top row of Figure 3 shows the results of the color histogram based tracker. The tracker totally loses the target after several frames. The bottom row in Figure 3 shows the tracking results using our approach. We can see that the tracker robustly follows the target despite confusing background color.

In particle filter tracking, the robustness and the tracking speed are proportional to the number of samples used. A large number of samples provides more robust results at the price of high computational load and slow tracking speed. Experiments show that the proposed tracker can successfully track the target with a reduced number of particles. For example, our method robustly follows the target in the Plane sequence with as few as only 20 particles, which significantly reduces the computational time.

In the second experiment, the Face sequence is used to evaluate the performance of the proposed tracker against severe and sudden illumination changes. The results are shown in Figure 4. It is important to point out that because of the poor lighting environment, the background color distribution is similar to that of the face. This makes the color based tracker to lose the target. Furthermore, the color based tracker can hardly adapt to a sudden illumination change. On the contrary, the proposed saliency based tracker performs extremely well in this situation. Note that the tracker is also robust against occlusion since it can recover the target object even if it is fully occluded as shown by frames 612 and 679 in Figure 4.

For a quantitative evaluation of the tracking methods, we use the spatial overlap metric defined in [13]. Let $S_{GT}$ and $S_T$ be, respectively, the ground truth and the estimated bounding box of the object in the $i$-th frame of a sequence. The spatial overlap is defined as:

$$\zeta_i = \frac{\text{Area}(S_{GT} \cap S_T)}{\text{Area}(S_{GT} \cup S_T)}. \tag{11}$$

The object in frame $k$ is accurately estimated if $\zeta_i > T$, where the threshold $T$ is set to 0.25 in our experiments.

The tracking performances are given in Table 1. For each method, the accuracy is defined as the ratio between the number of frames the object location is accurately estimated and the total number of frames in the sequence. Our saliency based tracker outperforms the color based tracker for all three sequences, providing excellent results for the Plane and Face sequences. In the case of the Walk sequence, the tracking accuracy is limited due to the fact that the walking person (the target) is hardly distinguishable from the background. However, using visual saliency improves the tracking results as shown in Figure 5.
Figure 2: Similarity measure between color and saliency distributions. (a) Reference color and saliency distributions. (b) First color and saliency candidate distributions. (c) Second color and saliency candidate distributions. Using color feature, the distances between the color distributions are respectively, \( \rho_{12} = 0.4983 \) and \( \rho_{13} = 0.4933 \). Using saliency distributions, the distances are \( \rho'_{12} = 0.1963 \) and \( \rho'_{13} = 0.3620 \).

Figure 3: Tracking results using the Plane sequence. Top row shows results with the color based tracker and bottom row shows results with the proposed method.

Figure 4: Tracking results in the presence of severe illumination change and occlusion. The target face is consistently and robustly tracked by the proposed method despite poor lighting condition and occlusion.
Figure 5: Tracking results using the Walk sequence. Top row shows results with the color based tracker and bottom row shows results with the proposed method.

<table>
<thead>
<tr>
<th>sequence</th>
<th>correct/total</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane</td>
<td>201/300</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>300/300</td>
<td>100</td>
</tr>
<tr>
<td>Face</td>
<td>429/680</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>671/680</td>
<td>98</td>
</tr>
<tr>
<td>Wall</td>
<td>21/70</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>48/70</td>
<td>68</td>
</tr>
</tbody>
</table>

C = color based tracker
C-S = color and saliency based tracker.

5. CONCLUSIONS

In this paper a robust tracking method is proposed. It is based on the integration of visual saliency information into the particle filter framework. We have shown how to effectively combine color and saliency information in order to make the tracker robust against occlusion, confusing background color and large illumination variation. Experiments with different sequences show that the proposed tracking method outperforms the established color based tracker, while requiring a reduced number of particles. A direction of future work would be an extension for multi-object tracking, incorporating shape and texture features.

REFERENCES


