A Framework to Integrate Particle Filters for Robust Tracking in Non-Stationary Environments

Francesc Moreno-Noguer and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, UPC-CSIC
Llorens Artigas 4-6, 08028, Barcelona, Spain
{fmoreno,asanfeliu}@iri.upc.es

Abstract. In this paper we propose a new framework to integrate several particle filters, in order to obtain a robust tracking system able to cope with abrupt changes of illumination and position of the target. The proposed method is analytically justified and allows to build a tracking procedure that adapts online and simultaneously the colorspace where the image points are represented, the color distributions of the object and background and the contour of the object.

1 Introduction

The integration of several visual features has been commonly used to improve the performance of tracking algorithms [1, 3, 9, 10]. However, all these methods lack a robust dynamic model to track the state of the features and cope with abrupt and unexpected changes of the target's position or appearance. Particle filters have been demonstrated to be robust enough to track complex dynamics. Usually, particle filters have been applied to only one object feature. [4] tracks an object based on multiple hypotheses of its contour. Subsequently, several approaches [7, 8] predict the target position based on the particle filter formulation. In our previous work [6] we proposed the use of this framework to predict the object and background color distributions.

In this work, we introduce a framework for the integration of several particle filters which are not independent between them, so that we can fuse their respective predicted features. [5] integrates different particle filter algorithms for tracking tasks, but with the assumption that the algorithms are conditionally independent. That is, if particle filter $\mathcal{PF}_1$ is based on features $\mathbf{z}_1$ to estimate the state vector $\mathbf{x}_1$ and particle filter $\mathcal{PF}_2$ uses features $\mathbf{z}_2$ to estimate $\mathbf{x}_2$, for each whole state of the object $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$ it is assumed that, $p(\mathbf{z}_1, \mathbf{z}_2|\mathbf{X}) = p(\mathbf{z}_1|\mathbf{x}_1)p(\mathbf{z}_2|\mathbf{x}_2)$. But this assumption is very restrictive and many times is not satisfied. For instance, a usual method to weigh each one of the samples of a contour particle filter, is based on the ratio of the number of pixels inside the contour with object color versus the number of pixels outside the contour with background color. This means that the contour feature is not independent of the color feature. In this situation if $\mathbf{z}_1$ represents the color features and $\mathbf{z}_2$ the contour ones, the latter will be function of both $\mathbf{x}_1$ and $\mathbf{z}_1$, i.e. $\mathbf{z}_2 = \mathbf{z}_2(\mathbf{x}_1, \mathbf{z}_1)$. Previous equation should be rewritten as, $p(\mathbf{z}_1, \mathbf{z}_2|\mathbf{X}) = p(\mathbf{z}_1|\mathbf{x}_1)p(\mathbf{z}_2|\mathbf{z}_1, \mathbf{x}_1, \mathbf{x}_2)$. In this paper we will design a system that verifies this relation of dependence between object features. The main contributions of the paper are the following:

1. Proposal of a framework to integrate several conditionally dependent particle filters.
2. There is no restriction in the number of particle filters that can be integrated.
3. Use the method to develop a robust tracking system that: **(a)** Adapts online the color space where image points are represented. **(b)** Adapts the distributions of the object and background colorpoints. **(c)** Accommodates the contour of the object.

All these features make our system capable to track objects in complex situations, like unexpected changes of the scene color, or abrupt and non-rigid movements of the target, as will be shown in the results Section.

In Section 2 we will introduce the mathematical framework and analytical justification of the method. The features that will be used to represent the object are described in Section 3. In Section 4 we will depict details about the sequential integration procedure for the real tracking. Results and conclusions will be given in Sections 5 and 6.

2 Mathematical framework

In the general case, let's describe the object being tracked by a set of F features, $\mathbf{z}_1, \ldots, \mathbf{z}_F$, that are sequentially conditional dependent, i.e. feature i depends on feature $i-1$. Each one of these features is associated to a state vector $\mathbf{x}_1, \ldots, \mathbf{x}_F$, which conditional a posteriori probability $p_1 = p(\mathbf{x}_1|\mathbf{z}_1), \ldots, p_F = p(\mathbf{x}_F|\mathbf{z}_F)$ is estimated using a corresponding particle filter $\mathcal{PF}_1, \ldots, \mathcal{PF}_F$. For the whole set of variables we assume that the dependence is only in one direction:

$$\{\mathbf{z}_k = \mathbf{z}_k(\mathbf{z}_i, \mathbf{x}_i), \mathbf{x}_k = \mathbf{x}_k(\mathbf{x}_i, \mathbf{z}_i)\} \Longleftrightarrow i < k \tag{1}$$

Considering this relation of dependence we can add extra terms to the a posteriori probability computed for each particle filter. In particular, the expression for the a posteriori probability computed by $\mathcal{PF}_i$ will be $p_i = p(\mathbf{x}_i|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{z}_1, \ldots, \mathbf{z}_i)$. Keeping this in mind, next we will prove that the whole a posteriori probability can be computed sequentially, as follows:

$$\begin{aligned} P = p(\mathbf{X}|\mathbf{Z}) &= p(\mathbf{x}_1, \ldots, \mathbf{x}_F|\mathbf{z}_1, \ldots, \mathbf{z}_F) \\ &= p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)\cdots p(\mathbf{x}_F|\mathbf{x}_1, \ldots, \mathbf{x}_{F-1}, \mathbf{z}_1, \ldots \mathbf{z}_F) = p_1 p_2 \cdots p_F \end{aligned} \tag{2}$$

Proof. We will prove this by induction, and applying Bayes' rule [2] and Eq. 1:

– Proof for 2 features:

$$p(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1, \mathbf{z}_2) = p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)p(\mathbf{x}_1|\mathbf{z}_1, \mathbf{z}_2) = p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$$

– For $F-1$ features we assume that

$$p(\mathbf{x}_1, .., \mathbf{x}_{F-1}|\mathbf{z}_1, .., \mathbf{z}_{F-1}) = p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)\cdots p(\mathbf{x}_{F-1}|\mathbf{x}_1, .., \mathbf{x}_{F-2}, \mathbf{z}_1, ..\mathbf{z}_{F-1}) \tag{3}$$

– Proof for F features:

$$\begin{aligned} p(\mathbf{x}_1, .., \mathbf{x}_F|\mathbf{z}_1, .., \mathbf{z}_F) &= p(\mathbf{x}_F|\mathbf{x}_1, .., \mathbf{x}_{F-1}, \mathbf{z}_1, .., \mathbf{z}_F)p(\mathbf{x}_1, .., \mathbf{x}_{F-1}|\mathbf{z}_1, .., \mathbf{z}_{F-1}) \\ &\overset{Eq.\ 3}{=} p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)\ldots p(\mathbf{x}_F|\mathbf{x}_1, \ldots, \mathbf{x}_{F-1}, \mathbf{z}_1, \ldots \mathbf{z}_F) \end{aligned}$$

Eq.2 tells us that the whole a posteriori probability density function can be computed sequentially, starting with $\mathcal{PF}_1$ to generate $p(\mathbf{x}_1|\mathbf{z}_1)$ and use this to estimate $p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$ with $\mathcal{PF}_2$, and so on.

In the iterative performance of the method, $\mathcal{PF}_i$ also receives as input at iteration t, the output pdf of its state vector $\mathbf{x}_i$ at the iteration $t-1$. We write the time expanded version of the *pdf* for $\mathcal{PF}_i$ as $p_i^{(t)} = p(\mathbf{x}_i^{(t)}|\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_{i-1}^{(t)}, \mathbf{z}_1^{(t)}, \ldots, \mathbf{z}_i^{(t)}, p_i^{(t-1)})$. We can also expand the expression of the whole *pdf* from Eq.2 as follows:

$$
\begin{aligned}
P^{(t)} &= p(\mathbf{X}^{(t)}|\mathbf{Z}^{(t)}) = p(\mathbf{x}_1^{(t)}, .., \mathbf{x}_F^{(t)}|\mathbf{z}_1^{(t)}, .., \mathbf{z}_F^{(t)}) \\
&= p(\mathbf{x}_1^{(t)}|\mathbf{z}_1^{(t)}, p_1^{(t-1)}) \cdots p(\mathbf{x}_F^{(t)}|\mathbf{x}_1^{(t)}, .., \mathbf{x}_{F-1}^{(t)}, \mathbf{z}_1^{(t)}, ..\mathbf{z}_F^{(t)}, p_F^{(t-1)}) = p_1^{(t)} p_2^{(t)} \cdots p_F^{(t)}
\end{aligned}
$$

Now let's describe in some detail the updating procedure of the $i-th$ particle filter, $\mathcal{PF}_i$. At time t, the filter receives $p_i^{(t-1)}$, the *pdf* of the state vector $\mathbf{x}_i$ at time $t-1$. This distribution is approximated by a set of samples $\mathbf{s}_{ij}^{(t-1)}$, $j = 1 \ldots N_i$, with associated weights $\pi_{ij}^{(t-1)}$. Given the set $\{\mathbf{s}_{ij}^{(t-1)}, \pi_{ij}^{(t-1)}\}$ the value of $p_i^{(t)}$ is estimated using the standard particle filter procedure:

1. The set $\{\mathbf{s}_{ij}^{(t-1)}, \pi_{ij}^{(t-1)}\}$, $j = 1 \ldots N_i$ is resampled (sampling with replacement) according to the weights $\pi_{ij}^{(t-1)}$. We obtain the new set $\{\mathbf{s'}_{ij}^{(t-1)}, \pi_{ij}^{(t-1)}\}$.
2. Particles $\mathbf{s'}_{ij}^{(t-1)}$ are propagated to the new set $\{\mathbf{s}_{ij}^{(t)}\}$, $j = 1 \ldots N_i$, based on the random dynamic model $\mathbf{s}_{i,j}^{(t)} = \mathcal{H}_i \mathbf{s'}_{i,j}^{(t-1)} + \mathbf{p}_i$, where $\mathcal{H}_i \sim \mathcal{A}_{3\times3}(0, \sigma_{H_i})$ and $\mathbf{p}_i \sim \mathcal{T}_{3\times1}(\mu_{p_i}, \sigma_{p_i})$. We define the matrix $\mathcal{A}$ and the vector $\mathcal{T}$ as follows:

$$
\mathcal{A}_{m\times m}(\mu_A, \sigma_A) = \begin{bmatrix} 1 + a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & 1 + a_{mm} \end{bmatrix} \qquad \mathcal{T}_{m\times1}(\mu_t, \sigma_t) = [t_1, \ldots, t_m]^T \tag{4}
$$

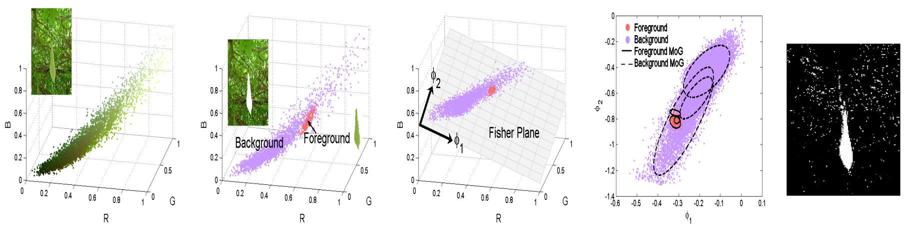
   where $a_{ij} \sim \mathcal{N}\left(\mu_{A_{ij}}, \sigma_{A_{ij}}\right)$, $t_i \sim \mathcal{N}\left(\mu_{t_i}, \sigma_{t_i}\right)$.
3. Finally, using some external measure on the feature $\mathbf{z}_i^{(t)}$ (updated with the values of the set of features $\{\mathbf{z}_k^{(t)}\}, k < i$ and its corresponding state vectors $\{\mathbf{x}_k^{(t)}\}$) , samples $\mathbf{s}_{ij}^{(t)}$ are weighted in order to obtain the output of iteration t, that is $\{\mathbf{s}_{ij}^{(t)}, \pi_{ij}^{(t)}\}$, $j = 1 \ldots N_i$, approximating $p_i^{(t)}$.

3 Features used for a robust tracking

In order to design a system able to work in real and dynamic environments we define a set of features that include both appearance (normal direction of the Fisher plane [6] and the color distribution of the object) and geometric attributes (contour) of the object. Next we will describe each one of these features:

3.1 Normal to the Fisher plane

In [6] we first introduced the concept of Fisher colorspace, and suggested that for tracking purposes the best colorspace is one that maximizes the distance between the object and background colorpoints. Let the sets $\mathcal{C}_{\mathcal{O}}^{RGB} = \{\mathbf{c}_{\mathcal{O},i}^{RGB}\}$, $i = 1, \ldots, N_{\mathcal{O}}$ and



(a) (b) (c) (d) (e)

Fig. 1. Color model. (a) All image points in the RGB colorspace. In the upper left part the original image is shown. (b) Manual classification of image points in foreground ($\mathcal{O}$) and background ($\mathcal{B}$). (c) Projection of $\mathcal{O}$ and $\mathcal{B}$ points on the Fisher plane. (d) *MoG* of $\mathcal{O}$ (the central leave) and $\mathcal{B}$ in the Fisher colorspace. (e) $p\left(\mathcal{O}|\mathbf{c}^{Fisher}\right)$, where brighter points correspond to more likely pixels.

$\mathcal{C}_{\mathcal{B}}^{RGB} = \{\mathbf{c}_{\mathcal{B},j}^{RGB}\}$, $j = 1, \ldots, N_{\mathcal{B}}$ be the colorpoints of the object and background respectively, represented in the 3-dimensional RGB colorspace.

Fisher plane $\Phi = [\phi_1, \phi_2] \in \mathcal{M}_{3\times 2}$ is computed applying the nonparametric Linear Discriminant Analysis technique [2] over the sets $\mathcal{C}_{\mathcal{O}}^{RGB}$ and $\mathcal{C}_{\mathcal{B}}^{RGB}$. An RGB colorpoint $\mathbf{c}^{RGB}$ is transformed to the 2D Fisher colorspace by $\mathbf{c}^{Fisher} = \Phi^T \mathbf{c}^{RGB}$ (see Fig. 1). This colorspace is adapted online, through the particle filter formulation presented above, with a 3D state vector corresponding to its normal vector, $\mathbf{x}_1 = \phi_1 \times \phi_2$.

3.2 Color distribution of the foreground and background

In order to represent the color distribution of the foreground and background in the Fisher colorspace, we use a *mixture of gaussians* (*MoG*) model. The conditional probability for a pixel $\mathbf{c}^{Fisher}$ belonging to a multi-colored object $\mathcal{O}$ is expressed as a sum of M_o gaussian components: $p\left(\mathbf{c}^{Fisher}|\mathcal{O}\right) = \sum_{j=1}^{M_o} p\left(\mathbf{c}^{Fisher}|j\right) P(j)$. Similarly, the background color will be represented by a mixture of M_b gaussians. Given the foreground ($\mathcal{O}$) and background ($\mathcal{B}$) classes, the a posteriori probability that a pixel $\mathbf{c}^{Fisher}$ belongs to object $\mathcal{O}$ is computed using the Bayes rule (Fig. 1d,e):

$$p\left(\mathcal{O}|\mathbf{c}^{Fisher}\right) = \frac{p\left(\mathbf{c}^{Fisher}|\mathcal{O}\right) P(\mathcal{O})}{p\left(\mathbf{c}^{Fisher}|\mathcal{O}\right) P(\mathcal{O}) + p\left(\mathbf{c}^{Fisher}|\mathcal{B}\right) P(\mathcal{B})} \tag{5}$$

where $P(\mathcal{O})$, $P(\mathcal{B})$ are the a priori probabilities of $\mathcal{O}$ and $\mathcal{B}$.

The configurations of the *MoG* for $\mathcal{O}$ and $\mathcal{B}$ are parameterized by the vector $\mathcal{G}_{\varepsilon} = [\mathbf{p}_{\varepsilon}, \mu_{\varepsilon}, \lambda_{\varepsilon}, \theta_{\varepsilon}]$ where $\varepsilon = \{\mathcal{O}, \mathcal{B}\}$, $\mathbf{p}_{\varepsilon}$ contains the priors for each gaussian component, μ_{ε} the centroids, λ_{ε} the eigenvalues of the principal directions and θ_{ε} the angles between the principal directions and the horizontal. $\mathbf{x}_2 = \{\mathcal{G}_{\mathcal{O}}, \mathcal{G}_{\mathcal{B}}\}$ will be the state vector representing the color model.

3.3 Contour of the object

Since color segmentation usually gives a rough estimation about the object location, we use the contour of the object, to obtain a more precise tracking. The contour will be represented by N_c points in the image, $\mathbf{r} = [(u_1, v_1), \ldots, (u_{N_c}, v_{N_c})]^T$. We assign these values to the state vector, $\mathbf{x}_3 = \mathbf{r}$.

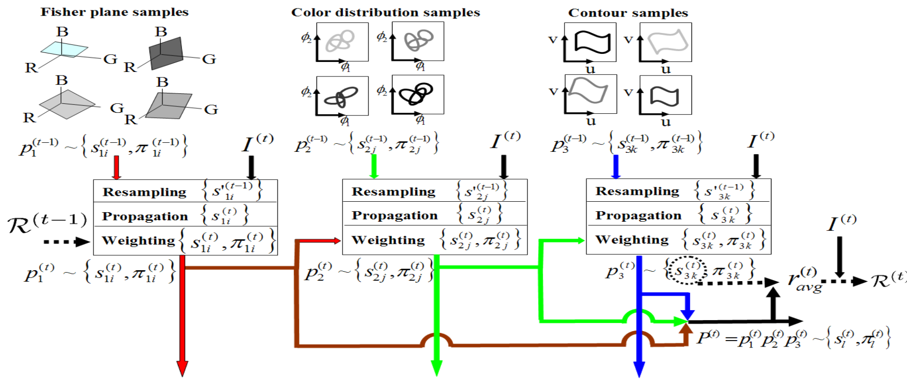


Fig. 2. Flow diagram of one iteration of the complete algorithm.

4 The complete tracking algorithm

In this Section we will integrate the tools described previously and analyze the complete method for tracking rigid and non-rigid objects in cluttered environments, under changing illumination. Let's describe the algorithm step by step (See Fig. 2):

4.1 Input at iteration t

At time t, for each i-feature, $i = 1, \ldots, 3$, a set of N_i samples $\mathbf{s}_{ij}^{(t-1)}$, $j = 1, \ldots, N_i$ (with the same structure than $\mathbf{x}_i$), is available from the previous iteration. Each sample has an associated weight $\pi_{ij}^{(t-1)}$. The whole set represents an approximation the a posteriori *pdf* of the system, $P^{(t-1)} = p(\mathbf{X}^{(t-1)}|\mathbf{Z}^{(t-1)})$, where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ contains the state vectors, and $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ refers to the measured features. Also available is the set of image points $\mathcal{R}^{(t-1)}$ that discretizes the contour of the object, and the input RGB image at time t, $I^{RGB,(t)}$.

4.2 Updating the Fisher plane *pdf*

At the starting point of iteration t, $\mathcal{PF}_1$, the particle filter associated to $\mathbf{x}_1$, receives at its input $p_1^{(t-1)}$, the *pdf* of the state vector $\mathbf{x}_1$ at time $t-1$, approximated with N_1 weighted samples $\{\mathbf{s}_{1j}^{(t-1)}, \pi_{1j}^{(t-1)}\}$, $j = 1, \ldots, N_1$. These particles are resampled and propagated to the set $\{\mathbf{s}_{1j}^{(t)}\}$ according to the dynamic model. Each sample represents a different Fisher plane, Φ_j, $j = 1, \ldots, N_1$. In order to assign a weight to each propagated sample, we define a region W in the image $I^{RGB,(t)}$, where we expect the object will be (bounding box around the contour $\mathcal{R}^{(t-1)}$). We fit a *MoG* configuration to the points inside and outside W, and assign a weight to each Fisher plane Φ_j depending on how well it discriminates the two regions:

$$\pi_{1j}^{(t)} \sim \frac{1}{N_W} \sum_{(u,v)\in W} p\left(W|I(u,v)_j^{Fisher,(t)}\right) - \frac{1}{N_{\overline{W}}} \sum_{(u,v)\notin W} p\left(W|I(u,v)_j^{Fisher,(t)}\right) \quad (6)$$

where $I_j^{Fisher,(t)}$ is the image $I^{RGB,(t)}$ projected on the plane Φ_j, and N_W, $N_{\overline{W}}$ are the number of image pixels in and out of W, respectively.

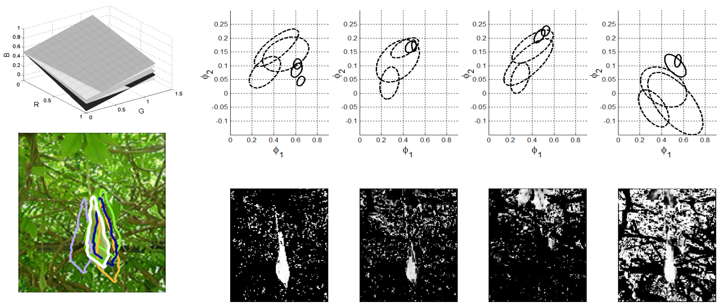


Fig. 3. Generation of multiple hypotheses for each feature. Upper left: Fisher plane. Lower left: Contour of the object. Right: Color distributions (and the corresponding a posteriori *pdf*s maps).

4.3 Updating the foreground and background color distributions *pdf*'s

$\mathcal{PF}_2$, the particle filter associated to the state vector $\mathbf{x}_2$, receives at its input $p_2^{(t-1)} \sim \{\mathbf{s}_{2j}^{(t-1)}, \pi_{2j}^{(t-1)}\}$, $j = 1, \dots, N_2$, approximating the *pdf* of the color distributions in the previous iteration, and $p_1^{(t)} \sim \{\mathbf{s}_{1k}^{(t)}, \pi_{1k}^{(t)}\}$, $k = 1, \dots, N_1$, an approximation to the *pdf* of the Fisher planes at time t. Particles $\{\mathbf{s}_{2j}^{(t-1)}\}$ are resampled and propagated (using the dynamic model associated to $\mathbf{x}_2$) to the set $\{\mathbf{s}_{2j}^{(t)}\}$. A sample $\mathbf{s}_{2j}^{(t)}$ represents a *MoG* configuration for the foreground and background. For the weighting stage, we associate to this sample, a sample of Fisher plane from $\mathcal{PF}_1$, in such a way that those samples $\mathbf{s}_{1k}^{(t)}$ of Fisher planes having higher probabilities will be assigned more times to the samples $\mathbf{s}_{2j}^{(t)}$ of *MoG*s. The weighting function is similar as before, but now the *MoG*s are provided by the sample $\mathbf{s}_{2j}^{(t)}$.

$$\pi_{2j}^{(t)} \sim \frac{1}{N_W} \sum_{(u,v)\in W} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) - \frac{1}{N_{\overline{W}}} \sum_{(u,v)\notin W} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) \quad (7)$$

4.4 Updating the contour *pdf*

$\mathcal{PF}_3$, receives at its input $p_3^{(t-1)} \sim \{\mathbf{s}_{3j}^{(t-1)}, \pi_{3j}^{(t-1)}\}$, $j = 1, \dots, N_3$, that approximates the *pdf* of the contours in the previous iteration, and $p_2^{(t)} \sim \{\mathbf{s}_{2k}^{(t)}, \pi_{2k}^{(t)}\}$, $k = 1, \dots, N_2$, an approximation to the *pdf* of the color distributions of foreground and background at time t. The set $\{\mathbf{s}_{3j}^{(t)}\}$ (the resampled and propagated particles, see Fig. 3) are weighted based on $p_2^{(t)}$ through a similar process than described for $\mathcal{PF}_2$: first we associate a sample $\mathbf{s}_{2k}^{(t)}$ to each sample $\mathbf{s}_{3j}^{(t)}$, according to the weight $\pi_{2k}^{(t)}$. Then we use the a posteriori probability map $p(\mathcal{O}|I_j^{Fisher,(t)})$ assigned to $\mathbf{s}_{2k}^{(t)}$ in the previous step, and the contour $\mathbf{r}_j$ represented by $\mathbf{s}_{3j}^{(t)}$ to compute the weight as follows:

$$\pi_{3j}^{(t)} \sim \frac{1}{N_{\mathbf{r}_j}} \sum_{(u,v)\in \mathbf{r}_j} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) - \frac{1}{N_{\overline{\mathbf{r}_j}}} \sum_{(u,v)\notin \mathbf{r}_j} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) \quad (8)$$

where $N_{\mathbf{r}_j}$ and $N_{\overline{\mathbf{r}_j}}$ are the number of image pixels inside and outside the contour $\mathbf{r}_j$.

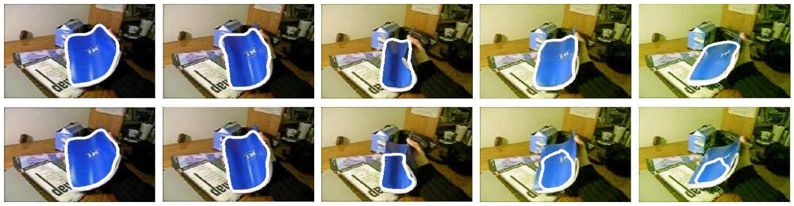


Fig. 4. Tracking results of a bending book in a sequence with smooth lighting changes. Upper row: using the proposed method the tracking works. Lower row: using only a contour particle filter and assuming smooth change of color the method fails.

The whole *pdf* can be approximated by a set of samples and weights:

$$P^{(t)} = P^{(t)}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = p_1^{(t)} p_2^{(t)} p_3^{(t)} \sim \{s_l^{(t)}, \pi_l^{(t)}\} \quad l = 1, \ldots, N_3 \quad (9)$$

Considering these final weights, the output contour is computed as $\mathcal{R}^{(t)} = \sum_{l=1}^{N_3} s_{3l}^{(t)} \pi_l^{(t)}$.

5 Experimental results

In this Section we examine the robustness of our system to several changing conditions of the environment, in situation where other algorithms may fail. In the first experiment we track the boundary of a bending book in a video sequence, where the lighting conditions change smoothly from natural lighting to yellow lighting. The upper row of Fig. 4 shows some frames of the tracked results. The same video sequence is processed by a particle filter that only uses multihypotheses for the prediction of the contour feature, while the color is predicted using a smooth dynamic model. Lower row of Fig. 4 shows that this method is unable to track the contour of the object and cope with the effects of self-shadowing produced during the movement of the book.

In the second experiment we have tested the algorithm with a sequence of a moving leave. Although this is a challenging sequence because it is highly cluttered, the illumination changes abruptly and the target moves unpredictably, the tracking results using the proposed method are good. Upper images of Fig. 5 show some frames of the tracking results. We show also the distribution of the weights for the samples of each particle filter. Observe that during the abrupt change of illumination (between frames 41 and 42), there is a compression of these curves. This means that the number of samples predicted well has been reduced. Nevertheless, the difference of probability between these samples and the rest of the samples has increased meaning that in next iteration the new predictions will be centered on these 'good' particles. We can observe that for frame 43 the tracking has stabilized. On the other hand, the lower images of Fig. 5 show the inability to accommodate these abrupt changes using a contour particle filter with smooth color prediction.

6 Conclusions

In this paper we have presented a new technique to integrate different particle filters that are conditionally dependent. This framework has allowed us to design a tracking algorithm that accommodates simultaneously the colorspace where the image points

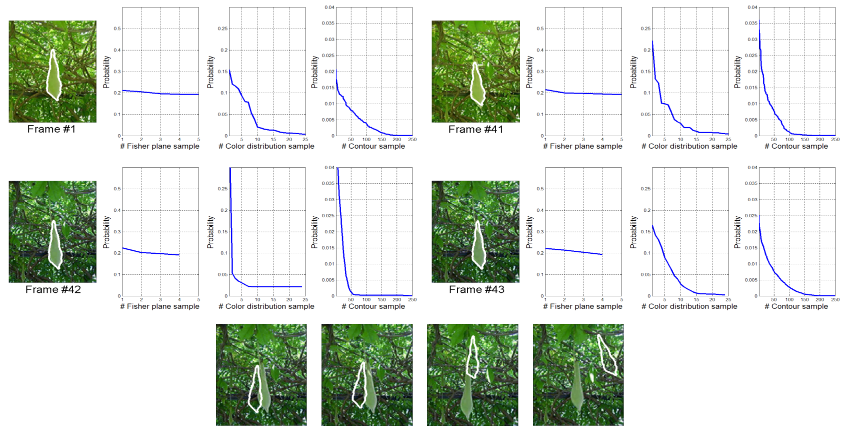


Fig. 5. Tracking results of a cluttered sequence with abrupt change of illumination and unpredictable movement of the target. Up: Results using the proposed method, and weight distribution for each particle filter. Down: Results assuming smooth change of color.

are represented, the color distributions of the object and background and the contour of the object. We have demonstrated the effectiveness of the method both analytically and experimentally, tracking real sequences presenting high content of clutter, non-rigid objects, non-expected target movements and abrupt changes of illumination.

References

1. S.Birchfield, “Elliptical head tracking using intensity gradients and color histograms”, *Proc. CVPR*, pp.232-237, 1998.
2. K.Fukunaga, “Introduction to statistical pattern recognition”, *2nd ed., Academic Press*, 1990.
3. E.Hayman, J.O.Eklundh, “Probabilistic and voting approaches to cue integration for figure-ground segmentation”, *Proc. ECCV*, pp.469-486, 2002.
4. M.Isard, A.Blake, “CONDENSATION-Conditional Density Propagation for visual tracking”, *IJCV*, Vol.29(1), pp.5-28, 1998.
5. I.Leichter, M.Lindenbaum, E.Rivlin, “A probabilistic framework for combining tracking algorithms”, *Proc. CVPR*, Vol.2, pp.445-451, 2004.
6. F.Moreno-Noguer, A.Sanfeliu, D.Samaras, “Fusion of a Multiple Hypotheses Color Model and Deformable Contours for Figure Ground Segmentation in Dynamic Environments”, *Proc. ANM, CVPRw*, 2004.
7. K.Nummiaro, E.Koller-Meier, L.Van Gool,“An adaptive color-based particle filter”, *IVC*, Vol.2(1), pp.99-110, 2003.
8. H.Sidenbladh, M.J.Black, D.J.Fleet, “Stochastic tracking of 3D human figures using 2D image motion”, *Proc. ECCV*, pp.702-718, 2000.
9. M.Spengler, B.Schiele, “Towards robust multi-cue integration for visual tracking”, *Machine Vision and Applications*, Vol. 14(1), pp. 50-58, 2003
10. J.Triesch, C.von der Malsburg, “Democratic integration: self-organized integration of adaptive cues”, *Neural Computation*, Vol.13(9), pp.2049-2074, 2001.

[0] This work was supported by CICYT projects DPI2001-2223 and DPI2000-1352-C02-01, and by a fellowship from the Spanish Ministry of Science and Technology.