Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 14, Iss. 2, pp. 4-15, 2016

Particle Filter and Visual Tracking: a Hybrid Resampling approach to improving robustness in cluttered and occluded environments

Diego A. L. Cordoba
Mechatronics Systems PhD Candidate - University of Brasilia - Brasilia - Brazil
dlegarda1@hotmail.com

Carla M. C. C. Koike
Department of Computer Science - University of Brasilia - Brasilia - Brazil
ckoike@cic.unb.br

Flavio de Barros Vidal
Department of Computer Science - University of Brasilia - Brasilia - Brazil
fbvidal@unb.br

Abstract – Occlusions and cluttered environments represent real challenges for visual tracking methods. In order to increase robustness in such situations this article presents a method for visual tracking using a Particle Filter with Hybrid Resampling. Our approach consists of using a particle filter to estimate the state of the tracked object, and both particles' inertia and update information are used in the resampling stage. The proposed method is tested using a public benchmark and the results are compared with other tracking algorithms. The results show that our approach performs better in cluttered environments, as well as in situations with total or partial occlusions.

Keywords – Visual tracking, Particle Filter, Hybrid resampling, Occlusions, Cluttered environments.

1 Introduction

Over the years, visual tracking has gained special attention in research areas of computer vision community as an important technique used in many applications (for example mobile, aerial and manipulator robots), both in structured and unstructured environments [1, 2]. The process of combining visual tracking and others techniques is widely spread nowadays, specially when dealing with challenging situations such as with total or partial occlusions and cluttered environments.

The aim of visual tracking is to detect a target and to determine its position and trajectory in a video sequence. Applications in this field are becoming very common [3–5], along with the evolution and lower costs of camera and computer technologies.

Visual tracking can be seen as a correspondence subproblem in vision-based motion analysis. The correspondence problem deals with determining the matching between elements of two frames in a sequence. It can, then, be applied for tracking purposes by determining the movement of an entire target region over a sequence of images. Due to the small spatial and temporal differences between consecutive frames, the correspondence problem can also be stated as the problem of estimating the apparent motion of the image brightness pattern.

The solution for the correspondence problem can roughly follow two strategies: differential methods and window-matching methods. Differential techniques use the concept of optical flow, that is generated based on the spatial and temporal variations of the whole image brightness. Methodologies for motion detection based on differential techniques can be modified to perform object tracking in a sequence of images [6].

One major drawback of these techniques is the need of numerical calculation of derivatives, which can be impracticable in circumstances where there is high level of noise, reduced number of frames or the effect of aliasing in the image acquisition process. Window-matching techniques [7] determine the degree of similarity among regions in sequential images, so that an object may be recognized and its position inferred in next frames. Window-matching techniques can be applied to image tracking as well as to other issues in computing vision (e.g. stereo image [8], image stitching [9]).

Occlusions and cluttered environments represent real challenges for visual tracking methods, because in these conditions the target can no longer be observed. Since obstacles may be treated as non-linearities, non-linear algorithms, such as particle filter, are proposed to overcome occlusions and cluttered environments in tracking. A Particle filter is one of many techniques that perform Recursive Bayesian Estimation, and it estimates recursively the posterior density function over a certain state space. Many recent approaches using Particle Filters for visual tracking can be found in [10–15] besides many other references in a extensive available literature.

In the case of visual tracking, the density function is a representation of the probability of the target position in the previous frame of an image sequence. The main idea of Particle Filters is to represent the *a posteriori* density function by a set of random samples with associated weights. These associated weights are obtained by a function that reaches its maximum in those samples near the object distinguished features. A major concern regarding Particle Filters is related to the situation where many of its samples drift to low posterior probability regions. The Resampling stage aims to move the set of particles back towards regions in state space with higher posterior probability.

When applying a particle filter to visual tracking, the occurrence of an occlusion can cause the deviation of the particles to the wrong region of the image. As a result, the resampling stage has a great impact on the robustness of the visual tracking algorithm employing particle filter.

In this paper, we propose the use of a particle filter in association with an Hybrid Resampling strategy as a method for robust and accurate response on different visual tracking scenarios.

In Section 2, the Particle Filter methods are introduced and discussed and Section 3 presents the hybrid resampling strategy. In Section 4, the proposed algorithm is applied to two types of visual tracking situations and the results are commented and discussed. This article is an improved and full version of [16].

2 Color Based Particle Filter

Particle filter is a powerful and flexible estimation technique for nonlinear applications. It is based on simulation and it is usually applied to estimate *Bayesian Models* where all variables are connected in a Markov Chain [17]. Its main idea is to obtain an approximate representation of the posterior probability density function using a subsequent set of random samples with associated weights.

Let $\{X^i_{0:k}, w^i_k\}^{N_s}_{i=1}$ be a measure that describes a random posterior probability density function (PDF) $p(X_{0:k}|Y_{1:k})$, where $(X^i_{0:k}, i = 0, ..., N_s)$ is a set of support points with associated weights $(w^i_k, i = 0, ..., N_s)$. The state vector $X_{0:k} = (X_j, j = 0, ..., k)$ is the set of all states at time k. The measure vector $Y_{1:k} = (Y_j, j = 1, ..., k)$ is the set of all measures, such as sensors reading, at time k. The weights are normalized by $\sum_{i=1}^{N_s} w^i = 1$ and obtained by Equation 1,

$$p(X_{0:k}|Y_{1:k}) \approx \sum_{i=1}^{N_s} w^i_k \delta(X_{0:k} - X^i_{0:k}). \tag{1}$$

The technique of Importance Sampling [18] ensures that we can build an estimator if each X^i_j and sample weights are calculated according to Eqs. 2 and 3, as follows:

$$X^i_j \propto q(X^i_k|X^i_{k-1}, Y^i_k), \tag{2}$$

$$w^i_k \propto w^i_{k-1} \frac{p(Y_k|X^i_k)p(X_k|X^i_{k-1})}{q(X^i_k|X^i_{k-1}, Y^i_k)}. \tag{3}$$

The distribution $q(X^i_k|X^i_{k-1}, Y^i_k)$ is called *importance density* and a good choice for this distribution can be defined as $q(X^i_k|X^i_{k-1}, Y^i_k) = p(X_k|X^i_{k-1})$. Therefore, Equation 3 can be reduced to:

$$w^i_k \propto w^i_{k-1} p(Y_k|X^i_k). \tag{4}$$

A common problem in this algorithm is the degeneration effect, as explained in [19], and in order to solve it, we can use an effective sample size ($\hat{N}_{eff}$) defined by:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} w^i_k}. \tag{5}$$

2.1 Resampling

The resampling process eliminates particles with small weights. These weak particles are replaced by others with higher weights, which defines another set of samples as a better representation for discretized $p(X_k|Y_k)$:

$$p(X_k|Y_{1:k}) \approx \sum_{i=1}^{Ns} w^i_k \delta(X_k - X^i_k). \tag{6}$$

The result of the resampling process is a new set of particles with uniform weight $1/N_s$.

2.2 Color Distribution Model

For visual tracking, a color-based model is used to increase robustness in situations with non-rigidity, rotation and partial occlusion in image domain. In our approach, we have choosen the HSV over the RGB color space, mainly due to its better stability under lighting changes. A descriptor based on color histogram (with 10 bins for Hue (H) and Saturation(S) channels) was used as input (measure states) for the proposed particle filtering scheme.

2.3 Weights setup

For each generated sample (i.e. frame) of the input image, the histogram of the tracked region of interest H_i is evaluated. Then, the Bhattacharyya Distance [20], $d_{H_O-H_i}$, between H_i and the histogram of the tracked object, H_O, is calculated as shown in the Eqs. 7 and 8.

$$MB = \sum_i \sqrt{H_i}\sqrt{S_i}, \tag{7}$$

$$d_{H_O-H_i} = \sqrt{1-MB}. \tag{8}$$

This value is used to calculate the weight of each particle according to Equation 9:

$$w^i = exp(-\lambda d^2_{H_O-H_i}), \tag{9}$$

where λ is equal to 20 (as detailed in [21]) and $d_{H_O-H_i}$ is the value of Bhattacharyya distance for the sample i. The above equation assures that if the samples have a high similarity with the target's histogram, the weights are adjusted to large values. If the similarity is small, the weights are reduced to small values.

2.4 Updating Model

To update the target model, we use the average weight of the particles that are close to the tracked object: If this value is above a fixed threshold, w_{min}, then the histogram of the tracked object is updated. Our approach avoids that undesired color values of the tracked region surroundings update wrongly the histogram values ($H_{O:k}$), as proposed by [22] and shown in Equation 10 below:

$$H_{O:k} = (1-\alpha-\beta-\gamma)H_{O:k-1} + (\alpha)H_{\Sigma w^i.X} + (\beta)H_{MP} + (\gamma)H_{O:k=0}. \tag{10}$$

This equation is actually a weigthed combination of information present in:

- the histogram at beginning of the tracking, $H_{O:k=0}$;
- the histogram at the time $k-1$, $H_{O:k-1}$;
- the histogram resulting from the weigthed particles, $H_{O:k-1}$;
- and the histogram of the heavier particle, H_{MP}.

The values of α, β and γ are chosen according to changes in the target estimation: α is the normalized weight of the particles estimation, β is proportional to the normalized weight of the highest particle value and γ is defined as 0.1. In our experiments, this value was chosen because it achieved the best results in proposed approach.

3 A Colour Based Particle Filter with Hybrid Resampling

According to [18], it has been show that the optimal proposal distribution is the one that can minimize the variance of the particle weights (see Equation 4). But it cannot be used efficiently in situations when occlusions and cluttered environments happen, due to the fact that we cannot evaluate the probability density function (PDF) before the particles are drawn. Many works (as [23], [24] and [25]) have focused their efforts on suboptimal approaches, because the transition prior based particle filter (generic particle filter) is easy to implement and has been widely used to solve problems in real-world scenarios, especially in visual tracking. Several versions of the generic particle filter, from the *Condensation* algorithm to the *Unscented* transformation, failed in complex tracking scenarios due to the sample impoverishment problems.

In order to overcome the problems related to occlusions and cluttered environments, we modified the basic structure of a particle filter based on Sequential Importance Resampling(SIR) [26], which is described in the following sections.

3.1 Resampling Stage

A modified version of the basic resampling approach proposed by [27] is detailed in Algorithm 1, and modified version of SIR Particle Filter algorithm is shown in Algorithm 2. The difference between the present algorithm and the standard SIR particle filters available in [26], is that we use information from particles state space dynamic to solve issues related to occlusions in visual tracking approaches.

The tracking process initiates with the assumption that there is no target occlusion and the proposed particle filter uses the discrete dynamic state model described in Equation 11:

$$\begin{aligned} X'_k &= X_{k-1} + r_k, \\ X''_k &= X'_{k-1} + I_{k-1}, \\ Y_k &= h(X_k, s_k). \end{aligned} \tag{11}$$

Algorithm 1: Resampling Algorithm

Data: $[\{X_k^{j*}, w_k^j, i^j\}_{j=1}^{N_s}]$
Result: $[\{X_k^i, w_k^i\}_{i=1}^{N_s}]$
Initialization PDF $c_1 = 0$;
for $i = 2 : N_s$ **do**
  Build PDF: $c_i = c_{i-1} + w_k^i$
end for
Random initialization: $u_1 \sim U[\frac{1}{N_s}]$
for $j = 1 : N_s$ **do**
  Move along the PDF: $u_j = u_1 + N_s^{-1}(j-1)$;
  while $u_j > c_i$ **do**
    $i = i + 1$
  end while
  Assign sample $X_k^{j*} = X_k^i$;
  Assign weight $w_k^j = N_s^{-1}$;
  Assign parent $i^j = i$;
end for

Algorithm 2: Particle Filter with Hybrid Resampling

Data: $[\{X_{i:k-1}, w_{i:k-1}\}_{j=1}^{N_s}, Y_k]$
Result: $[\{X_{i:k}, w_{i:k}\}_{i=1}^{N_s}]$
initialization;
for $i = 1 : N_s$ **do**
  $X_{i:k} \sim p(X_k | X_{i:k-1})$;
  Calculate $w_{i:k}^* = p(Y_k | X_{i:k})$ (Equation 9);
end for
for $i = 1 : N_s$ **do**
  if $1 < \frac{wi*}{wi}$ **then**
    $X = X^*$ (Equation 11);
    $w_i = w_i^*$;
  else
    $X = X_{k-1}$ (Equation 11);
    $w_i = w_{i:k-1}$;
  end if
end for
Calculate $\bar{v}$ (Equation 12);
for $j = 1 : N_s$ **do**
  Normalization $w_{i:k} = \frac{w_{i:k}}{\sum i=1^{N_s} w_{i:k}}$;
end for
Calculate $\hat{N}_{eff}$ (Equation 5);
if $\hat{N}_{eff} < N_{lim}$ **then**
  Resampling Algorithm 1;
end if
Update Histogram target (Equation 10);
Estimate $\hat{X}$ (Equation 18);

The state vector X_k estimates the position (vertical and horizontal) of the target in the image domain, obtained from a rectangle that encloses the tracked object at time instant k. The state vector X'_k and X''_k are derivatives (first and second order respectively) of the target position at time instant k. The random variables r_k and s_k are mutually independent, modeled by Gaussian functions, and they describe the process and measuring noises respectively. $h(.)$ is the measure state function. I_{k-1} is the inertial factor, responsible for providing the inertial movement of the samples and obtained from a Gaussian distribution weighted by the velocity of the particles.

Assuming that the tracked object velocity v between frames is uniform, it may be evaluated as,

$$v_i = (X'_{i,k} - X'_{i,k-1}). \tag{12}$$

The expressions of transition probabilities are defined by Eqs. 13, 14 and 15 respectively.

$$\hat{p}(X_k|Y_{1:k}) = argmax\{\pi(X_{i:k})\}, \tag{13}$$

$$p(X_k|X_{k-1}) = p(X_{k-1}|r_k, I), \tag{14}$$

$$p(Y_k|X_k) = N(Y_k|h(X_k), s_k). \tag{15}$$

$p(Y_k|X_k)$ is a Normal distribution ($N(.)$), and $\pi(X_{i:k})$ is the *a posteriori* distribution from state samples X between time instant i and k, defined by Equation 16 and restricted by Equation 17:

$$\pi(X) = p(\mathbf{Y}_k|\mathbf{X}_{i:k})\left[\Omega(.)\right], \tag{16}$$

$$\Omega(.) = \sum_{i=1}^{\lambda} p(\mathbf{X}'_k|\mathbf{X}'_{i:k-1}) + \sum_{i=\lambda}^{N_s} p(\mathbf{X}''_k|\mathbf{X}''_{i:k-1}). \tag{17}$$

where $1 < \lambda < N_s$ and N_s is the maximum number of particles in the filter.

When the target can not be observed in a frame (for example because of an occlusion), the state transition for each set of particles is modified. The sum in Equation 17 changes: from a normal distribution to an uniform distribution around the last estimation before the occlusion, as described in Equation 18,

$$X'_i = \hat{X}_{i:k-1} + r_k, \tag{18}$$

where the state vector $\hat{X}_{k-1}$ describes the state posterior at the time before occlusion and r_k shows the value evaluated from a uniform distribution $U(u|l_k, u_k)$. In this case, lower(l_k) and upper(u_k) limits change during the remaining frames. The state vector X'' is used for latest update of the estimated velocity state vector X', before occlusion. Following these constraints, an estimation of the object position is possible when total occlusion occurs and/or in high cluttered environments. When the missing target reappears, or the occlusion is over, the particles are updated by the state transition shown in Equation 11. The approach proposed here aims to deal with problematic issues related to occlusion and cluttered environment using only the color distribution on the histogram of the tracked object, as described in Section 2.2, in the update model. We propose that the update occurs only when the histogram values from the tracked object at time $k-1$ (before the weights update step) have lower similarity ($\hat{N}_{eff} < N_{lim}$) to the region around of the tracked object at time k (using spatio-temporal constraints as described in [28]). Which means that an occlusion or a target drift in cluttered environments can be detected. After this detection stage, the process of Hybrid Resampling begins as described by Algorithm 1.

4 Experimental Results

In order to evaluate the proposal method, we applied our approach to the *Bonn Benchmark on Tracking - Bobot* [1] for 2D visual tracking problems. The *Bobot* dataset includes several sequences with many types of tracking objects such as people, mugs, cube, etc. and a complete ground-truth (GT) with spatial positions (horizontal and vertical) and size (height and weight) of each tracked object. All available sequences have a spatial resolution of 320 × 250 pixels and frame rate of 25 fps. The proposed Particle Filter with Hybrid Resampling (PFHR), described in Section 3, uses up to 200 particles. We compared the proposed method with two other algorithms: (a) a deterministic algorithm based on template matching (WM); and (b) a basic implementation of the SIR Particle Filter (also setup up to 200 particles). All techniques are implemented using C++ Language and OpenCV library [29] and computational hardware is a laptop computer with *Core i5* processor with 4 GB of RAM memory, with Ubuntu Linux.

Our method was run in more than six image sequences scenarios from the above mentioned benchmark. Here, we present some of the results for three sequences from the *Bobot Database* (additional results can be seen in Table 1). The first sequence presents many characteristics of cluttered environments (Subsection 4.1), as the second shows an outdoor sequence where several objects present high similarity with the tracked object and total occlusion happens in more the one occasion (Subsection 4.2). The third sequence illustrates an indoor situation with many total occlusions of the tracked object and light changes along the sequence (Subsection 4.3).

[1] Available in http://www.iai.uni-bonn.de/∼kleind/tracking/

4.1 Cluttered environments sequence

This sequence presents many abrupt background changes, camera motion and scale transformation. Figure 1 shows some relevant frames from the sequence with cluttered environments.

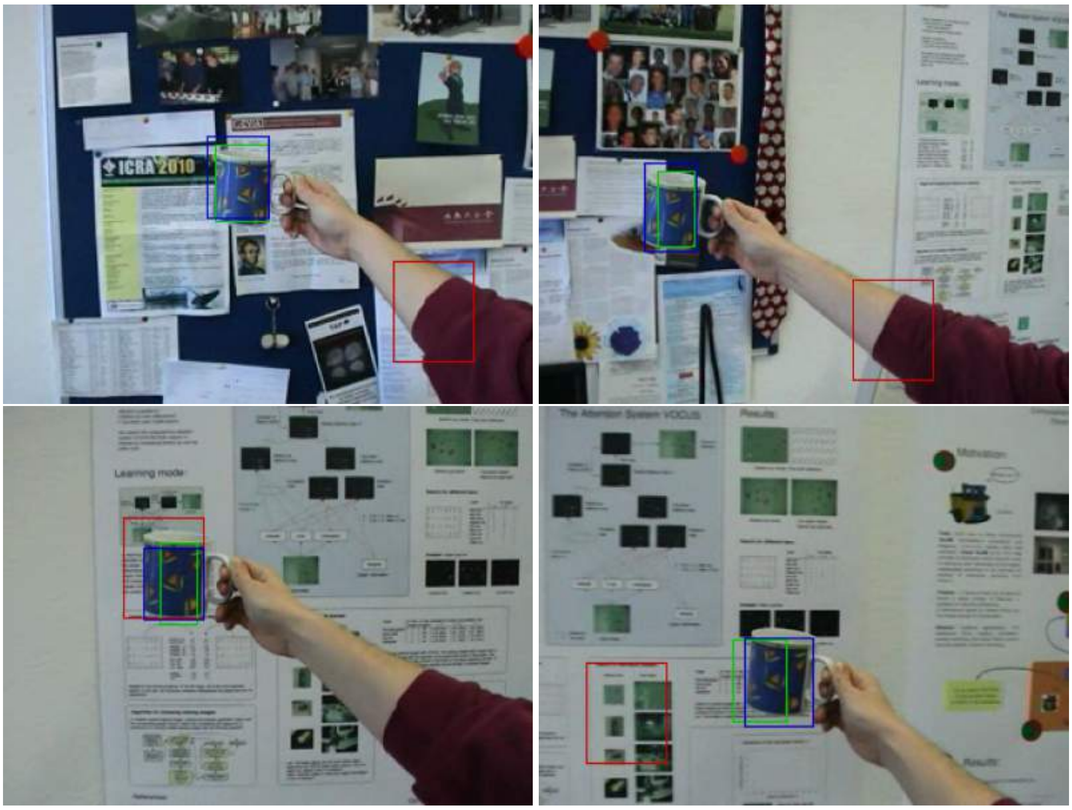
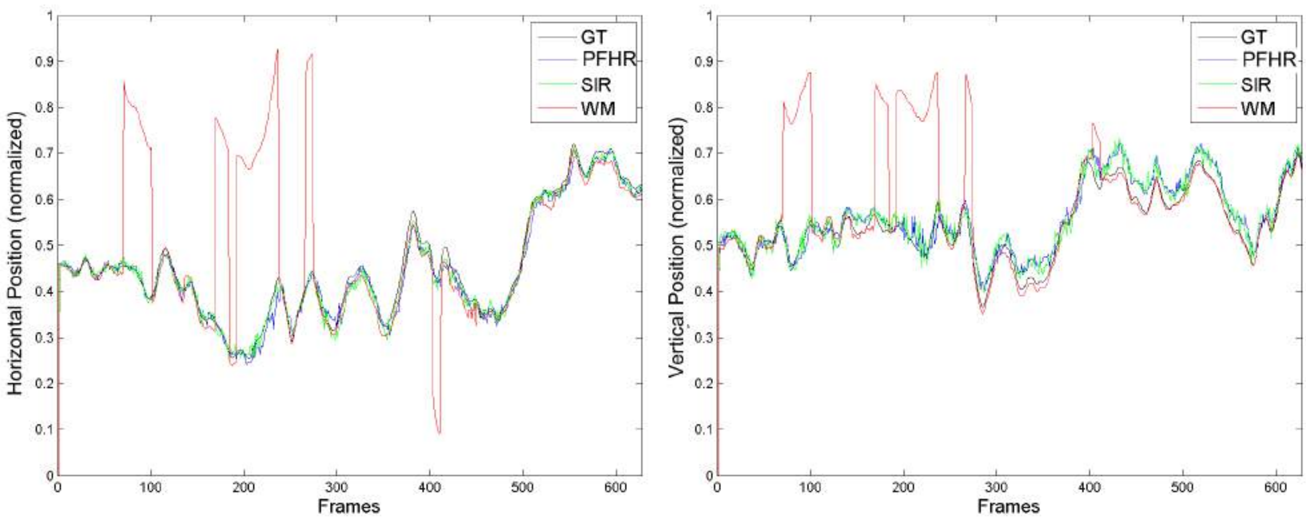


Figure 1: Tracking in cluttered environments. Legend: WM, SIR Particle Filter, PFHR.

Many errors occur in WM algorithm during the tracking of the blue mug (See Figs. 1 and 2), especially when similar objects appear at the bottom of the image. The basic SIR Particle Filter algorithm has an adequate response with respect to the spatial position of tracked object, but the object size is incorrectly estimated. On the other hand, PFHR does a much better job estimating both features, the position and the size of the tracked objects.



(a) (b)

Figure 2: Cluttered environment Sequence - Positions normalized Horizontal (a) and Vertical (b). Ground-truth (GT), WM, SIR Particle Filter, PFHR.

4.2 Outdoors with total occlusion sequence

This sequence shows a person (tracked object) walking outdoors while several others people cross the way, generating occlusions (Figure 3). Besides occlusions, the tracked object performs scale, rotation and translation changes during the sequence.

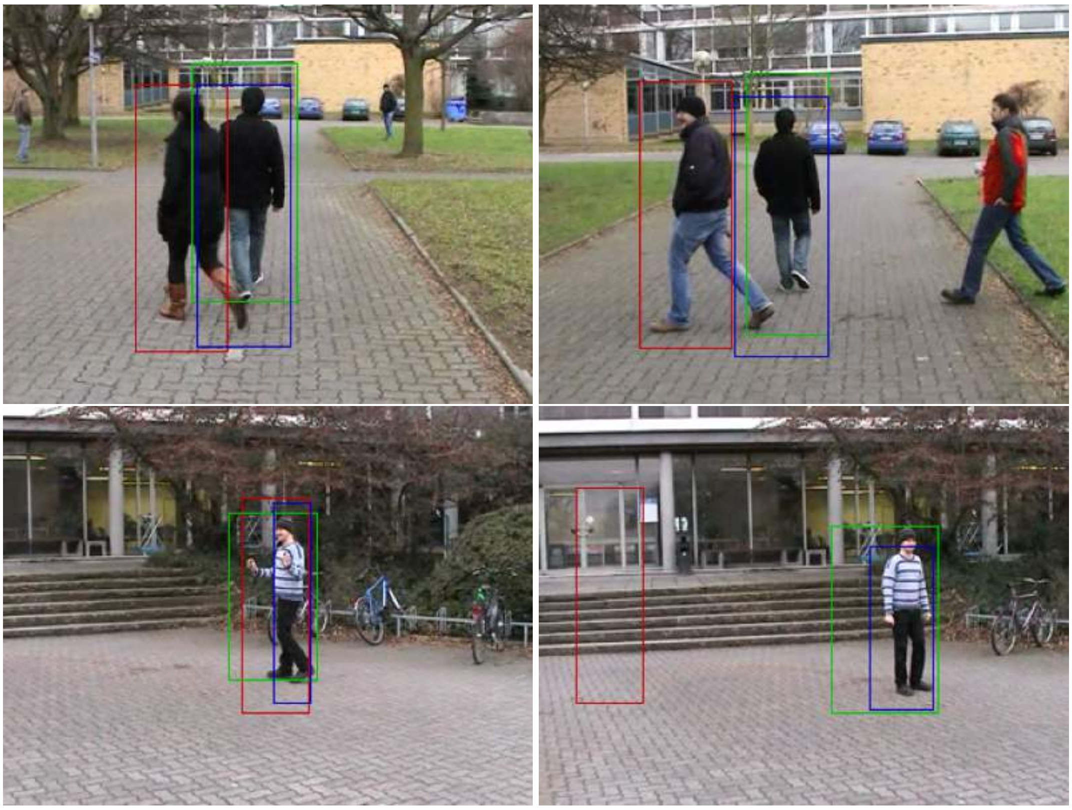


Figure 3: Sequence of the total occlusion of a person walking in outdoor environment. Legend: WM, SIR Particle Filter, PFHR.

The WM algorithm missed the tracked person after occlusion and in the presence of objects that are similar to the target[2]. The SIR Particle Filter also missed the target after occlusions and it was not able to properly estimate the size of the target. The PFHR algorithm was capable of estimating both the correct position and size of tracked person. When a persistent occlusion occurs, the PFHR strategy consists in evaluate *a posteriori* distribution of the state vector X from Equation 17, including values from the dynamic movement model in image domain (Equation 11). Specifically for this occlusion situation, we have set half of the number of particles to the first order model and second order model respectively. These values were chosen empirically to obtain the best result for the tracking performed.

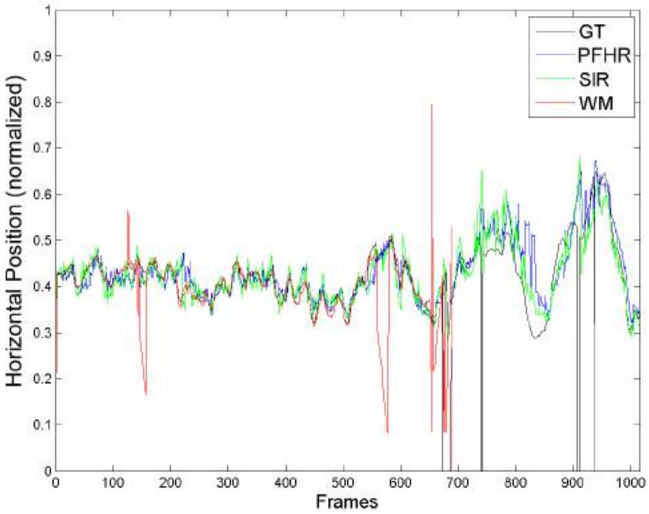
As can be seen in the sequence shown in Figure 3, WM is not able to track the person. The SIR Particle Filter algorithm can track correctly, but with large displacements positions (horizontal and vertical) variations along the frames, as shown in Figure 4. The PFHR track the person with small errors and, after occlusion occurs, with only small oscillations, the algorithm can recover the target after few frames.

4.3 Indoors with total occlusion sequence

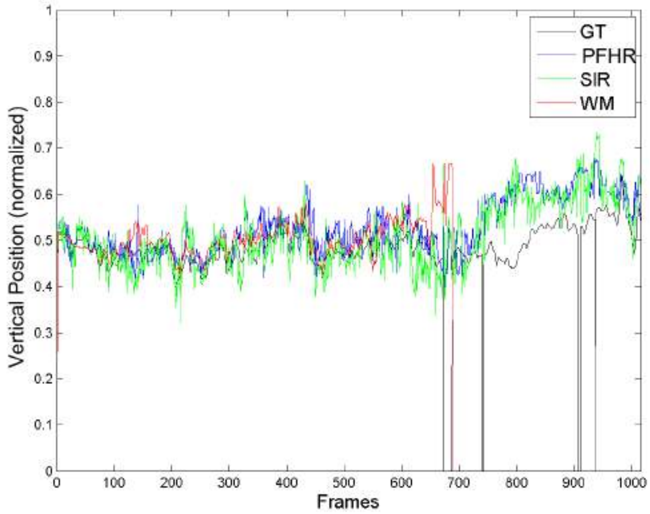
In this sequence, a man (tracked person) is walking through on indoor hall and the recording camera follows him up by his side. Due to the way the hall was built, as well as the position between target and camera, many target occlusions occurs caused by people and obstacles (Figure 5). The robustness of the proposal methodology is tested for several occlusions situations, demonstrating the versatility and stability of the proposed model of tracking as well the hybrid resampling strategy on the global tracking of the target process.

The WM algorithm missed the tracked person after occlusion, as shown in the Figures 6 - (a) and (b). The SIR Particle Filter also missed the target after occlusions and it was not able to keep tracking the target. The PFHR algorithm was the only technique able to estimate both the correct position and size of the tracked person. When a persistent occlusion occurs, the PFHR strategy evaluate the *a posteriori* distribution of the state vector X from Equation 17, including values from the dynamic movement model in image domain (Equation 11) and update the model dynamic inertia from the particle movement information. In this several occlusion situations, half of the number of particles were set to the first and second order models respectively, these values were chosen empirically to obtain the best result for the tracking performed.

[2]For Bobot Benchmark when an occlusion occurs the value assigned by the ground-truth to the horizontal and vertical positions is zero, respectively.

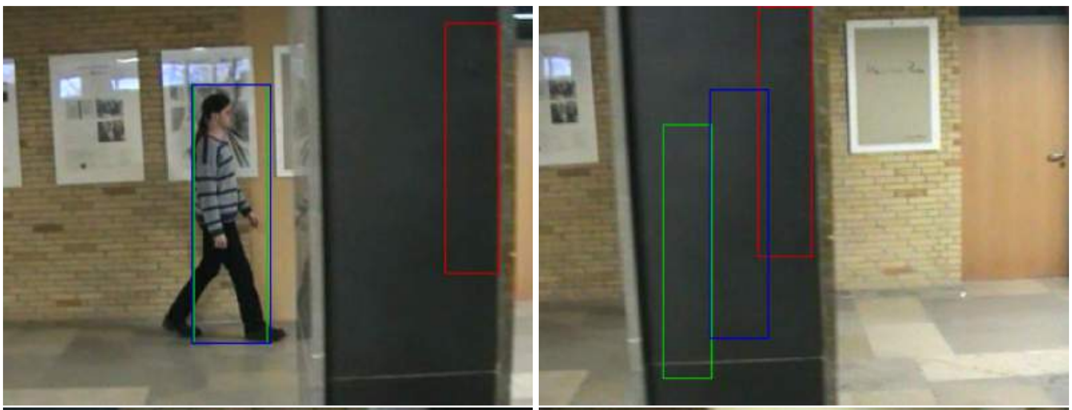


(a)



(b)

Figure 4: Outdoor occlusion sequence - Positions normalized Horizontal (a) and Vertical (b). Ground truth (GT), WM, SIR Particle Filter, PFHR.



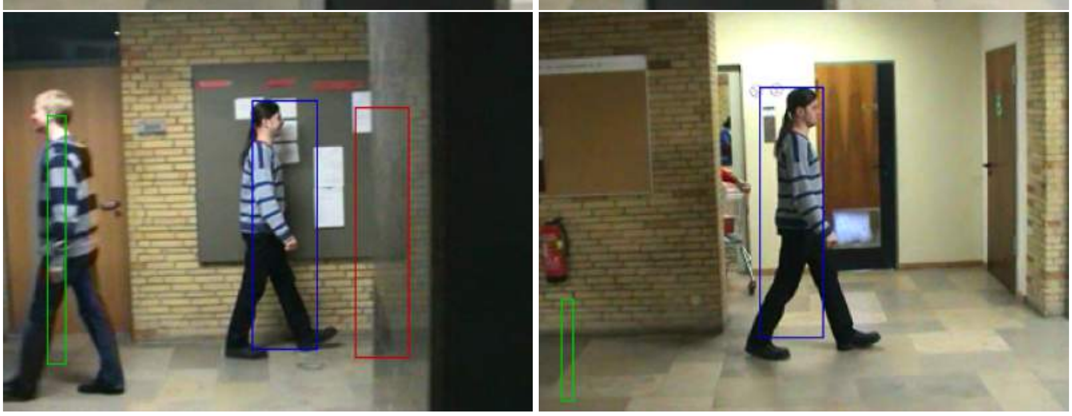
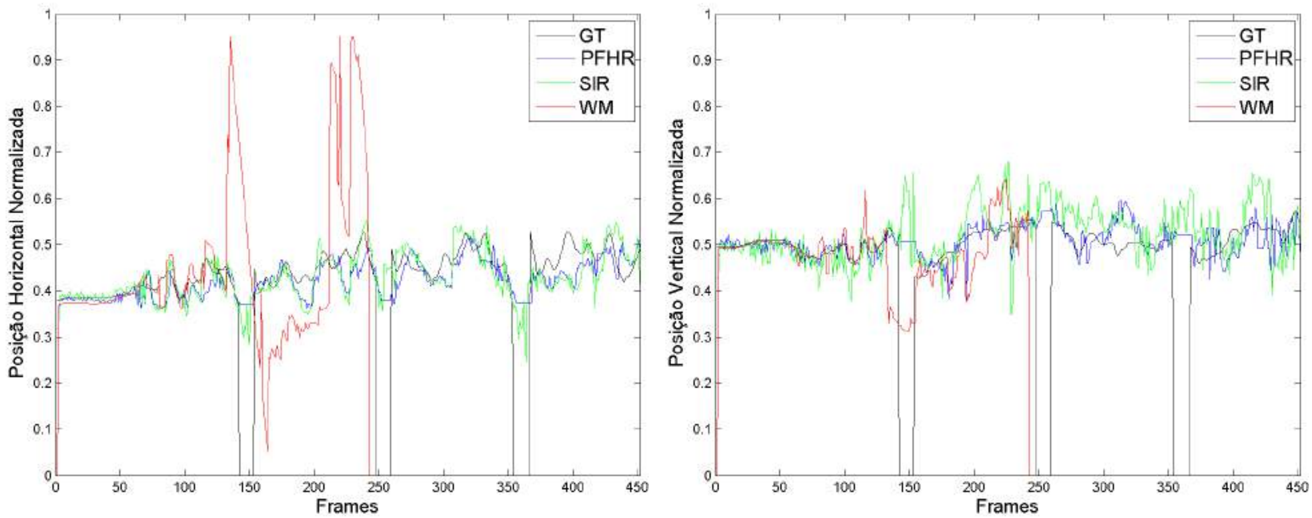


Figure 5: Sequence of the total occlusion of a person walking in indoor hall. Legend: WM, SIR Particle Filter, PFHR.

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 14, Iss. 2, pp. 4-15, 2016



(a) (b)

Figure 6: Indoor total occlusion sequence - Positions normalized Horizontal (a) and Vertical (b). Ground truth (GT), WM, SIR Particle Filter, PFHR.

4.4 Tracking Performance Evaluation

To adequately evaluate the tracking algorithm, the estimation of target size must also be taken into account, as proposed by [30]. It consists on measuring the overlap between ground truth and estimated areas, as defined by Equation 19:

$$A(GT, ST) = \frac{Area(GT \cap ST)}{Area(GT \cup ST)}, \tag{19}$$

where GT represents the *ground truth* area and ST represents the estimated area obtained by the tracking algorithm. In according to [30], if $A(GT, ST)$ is greater than threshold, T_{lim} (chosen to be at least 20%) then we have a true positive. Table 1 shows the results (in percentage) of true positives along of the total number of frames where the target is detected, for all method tests. It can be seen that PFHR shows a higher percentage than WM and SIR, for all tested sequences.

	PFHR	SIR	WM
Cluttered Environments	**97.60**	95.21	82.93
Outdoor w/ Total Occlusion	**93.79**	90.18	70.07
Background changes	**97.60**	95.21	82.93
Scale variations	92.30	**94.96**	33.90
Trajectory changes	**95.67**	76.87	87.18
Indoor w/ Total Occlusion	**88.93**	86.06	30.08
Overall Average	**94.75**	89.74	64.51

Table 1: Percentage of true positive data.

4.5 Computational complexity analysis

The evaluation of the computational complexity in [31] assigns for each line in the algorithm a variable that represents the running time, regardless the type of the variable (float, integer, ...). If the statement is inside a repetition structure (*while*, *for*, ...), the complexity increases proportionally to the nested loops. For the standard Particle Filter SIR (Section 4.5.1) considering n to be the standard number of input particles of the implemented Particle Filter, its computational complexity can be described as $O(n^2)$, with no relation to the number or size of samples. In the analysis of the computational complexity of the proposed algorithm, it is assumed that all lines have the same individual cost in relation to execution time.

4.5.1 Standard Particle Filter SIR - Complexity Analysis

1.	**Instruction**	**Comp. Cost**	**$\#$ of Iterations**
2.	**For loop** $i = 1 : N_s$	C_1	n

3.	$X_{i:k} \sim p(X_k \mid X_{i:k-1})$	C_2	$n-1$
4.	$w_i^* = P(Y_k \mid X_{i:k})$	C_3	$n-1$
5.	**For loop** $i = 1 : N_s$	C_4	n
6.	$t = \sum_i^{N_s} w_{i:k}$	C_5	$n-1$
7.	$N = \sum_i^{N_s} w_{i:k}^2$	C_6	$n-1$
8.	$N_{eff} = \frac{1}{N}$	C_7	1
9.	**Resampling Step**	C_8	n^2
10.	**For loop** $i = 1 : N_s$	C_9	n
11.	$\hat{X}_k = \sum_i^{N_s} w_{i:k} X_{i:k}$	C_{10}	$n-1$

In the above algorithm, C_c, with $c = \{1..10\}$, are coefficients of computational cost associated to time of execution of each line in the standard Particle Filter SIR Algorithm execution. The actual running time can differ, depending on the speed (frequency) of the choosen processor. In according to the above study, the resampling step is responsible for the quadratic term in the complexity, which defines the whole particle filter complexity to be $O(n^2)$

4.5.2 Hybrid Resampling Stage - Complexity Analysis

1.	**Instruction**	**Comp. Cost**	**# of Iteractions**
2.	**For loop** $i = 2 : N_s$	C_1	$n-1$
3.	$c_i = c_{i-1} + w_{i:k}$	C_2	$n-2$
4.	**For loop** $j = 1 : N_s$	C_3	n
5.	$u_j = u_1 + N_s^{-1}(j-1)$	C_4	$n-1$
6.	**While loop** $u_j > c_i$	C_5	$(n-1) \times n$
7.	$i = i + 1$	C_6	$(n-1) \times n$
8.	$X_j = X_i$	C_7	$n-1$

In Line 6 of the above algorithm, the Hybrid Resampling strategy includes a conditional loop and the worst situation assumes that u_j is higher than c_i in all loop iterations. It means that Hybrid Resampling Stage is always activated and the computational complexity is described as $O(n^2)$.

Comparing the computational complexity of the standard Particle Filter SIR and the algorithm here proposed (PFHR), the same computational complexity is achieved when the resampling methodology described in Algorithm 1 is applied.

5 Conclusions

In this work we presented a new approach for visual tracking, that uses a Particle Filter with Hybrid Resampling strategy in order to improve robustness. All tests show that the proposed algorithm (PFHR) achieve better results when compared to the classical techniques of visual tracking (WM and SIR), especially in occlusions and cluttered environments.

The presented approach would provide improvements for visual tracking due to the fact that the tracking is independent of the motion type (for example random trajectories) and of the object shape. The algorithm also offers flexibility in situations where there is no previous informations about the object to be tracked. This work follows in a continuous development and it will include the implementation of the proposed algorithm on a low level programming language in order to enable its operation in real time scenarios (including timing analysis). The next step is performing continuous comparisons with the latest techniques available in visual tracking literature to improve the proposed approach.

References

[1] I. Siradjuddin, L. Behera, T. McGinnity and S. Coleman. "Image-Based Visual Servoing of a 7-DOF Robot Manipulator Using an Adaptive Distributed Fuzzy PD Controller". *Mechatronics, IEEE/ASME Transactions on*, vol. PP, no. 99, pp. 1–12, 2013.

[2] F. Bakhshande and H. Taghirad. "Visual tracking in four degrees of freedom using kernel projected measurement". In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*, pp. 425–430, 2013.

[3] X. Gao, D. You and S. Katayama. "Seam Tracking Monitoring Based on Adaptive Kalman Filter Embedded Elman Neural Network During High-Power Fiber Laser Welding". *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 11, pp. 4315–4325, 2012.

[4] W. Ge, R. Collins and R. Ruback. "Vision-Based Analysis of Small Groups in Pedestrian Crowds". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 1003–1016, 2012.

[5] B. Leibe, K. Schindler, N. Cornelis and L. Van Gool. "Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 10, pp. 1683–1698, 2008.

[6] F. B. Vidal and V. H. C. Alcalde. "Motion segmentation in sequential images based on the differential optical flow". *2nd International Conference on Informatics in Control, Automation and Robotics -ICINCO*, pp. 94–100, 2005.

[7] P. Anandan. "A computational framework and an algorithm for the measurement of visual motion". In *International Journal of Computer Vision*, volume 2, pp. 283–310, 1989.

[8] E. Trucco and A. Verri. *Introductory Techiniques for 3-D Computer Vision*. 1998.

[9] G. Ward. "Hiding Seams in High Dynamic Range Panoramas". In *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, APGV '06, pp. 150–150, New York, NY, USA, 2006. ACM.

[10] L. Romo-Morales, A. Sanchez, V. Parra-Vega, O. Garcia and F. Ruiz-Sanchez. "Visual control for trajectory tracking of Quadrotors and real-time analysis on an emulated environment". In *American Control Conference (ACC), 2013*, pp. 6315–6321, 2013.

[11] W. Limprasert, A. Wallace and G. Michaelson. "Real-Time People Tracking in a Camera Network". *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 3, no. 2, pp. 263–271, 2013.

[12] K. Mohan and M. Wilscy. "Object ranging and tracking for aircraft landing system". In *Signal Processing Image Processing Pattern Recognition (ICSIPR), 2013 International Conference on*, pp. 278–282, 2013.

[13] X. Zhou, Y. Li and B. He. "Entropy distribution and coverage rate-based birth intensity estimation in GM-PHD filter for multi-target visual tracking". *Signal Processing*, vol. 94, no. 0, pp. 650 – 660, 2014.

[14] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidel and D. Stoyanov. "Optical techniques for 3D surface reconstruction in computer-assisted laparoscopic surgery". *Medical Image Analysis*, vol. 17, no. 8, pp. 974 – 996, 2013.

[15] T. Rui, Q. Zhang, Y. Zhou and J. Xing. "Object tracking using particle filter in the wavelet subspace". *Neurocomputing*, vol. 119, no. 0, pp. 125 – 130, 2013. Intelligent Processing Techniques for Semantic-based Image and Video Retrieval.

[16] F. de Barros Vidal, D. A. Cordoba, A. Zaghetto and C. M. Koike. "Improving visual tracking robustness in cluttered and occluded environments using particle filter with hybrid resampling". In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, pp. 605–612. SCITEPRESS, 2014.

[17] A. Doucet, D. Freitas, Nando, Gordon and Neil, editors. *Sequential Monte Carlo methods in practice*. 2001.

[18] A. Doucet, S. Godsill and C. Andrieu. "On sequential Monte Carlo methods for Bayesian filtering". *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[19] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on signal procesing*, vol. 50, 2002.

[20] O. Straka and M. Šimandl. *Using the Bhattacharyya distance in functional sampling density of particle filter*, pp. 1–6. IFAC, Prague, 2005.

[21] P. Perez, C. Hue, J. Vermaak and M. Gangnet. "Color-based probabilistic tracking". pp. 661–675, 2002.

[22] J. Li and C.-S. Chua. "Transductive inference for color-based particle filter tracking". vol. 3, pp. III–949–52 vol.2, 2003.

[23] N. Gordon, D. Salmond and A. F. M. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[24] R. van der Merwe, A. Doucet, N. de Freitas and E. A. Wan. "The Unscented Particle Filter". In *NIPS*, pp. 584–590, 2000.

[25] F. Wang, Y. Lin, T. Zhang and J. Liu. "Particle Filter with Hybrid Proposal Distribution for Nonlinear State Estimation." *JCP*, vol. 6, no. 11, pp. 2491–2501, 2011.

[26] N. Gordon, D. Salmond and A. F. M. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[27] G. Kitagawa. "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models". *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[28] B. P. K. Horn and B. G. Schunck. "Determining Optical Flow". *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.

[29] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[30] F. Yin, D. Makris and S. Velastin. "Performance Evaluation of Object Tracking Algorithms". *Proceeding Tenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[31] T. H. Cormen, C. Stein, R. L. Rivest and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, second edition, 2001.