Comparing a Kalman Filter and a Particle Filter in a Multiple Objects Tracking Application

M. Marrón, J.C. García, M.A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, J. Cerro
Electronics Department, University of Alcala, Alcalá de Henares, SPAIN

Abstract **— Two of the most important solutions in position estimation are compared, in this paper, in order to test their efficiency in a multi-tracking application in an unstructured and complex environment. A Particle Filter is extended and adapted with a clustering process in order to track a variable number of objects. The other approach is to use a Kalman Filter with an association algorithm for each of the objects to track. Both algorithms are described in the paper and the results obtained with their real-time execution in the mentioned application are shown. Finally interesting conclusions extracted from this comparison are remarked at the end.**

Keywords **– position estimation, probabilistic algorithms, robotics, tracking.**

I. Introduction

In any robotic and intelligent vehicle application the position estimation process is one of the most important tasks. This process estimation is needed both to extract information about the robot environment and to know the concrete location of the robot itself.

Taking into account the restrictions imposed by the measurements, the estimation process has to consider the noise related to them in order to achieve reliable information about the position. Thus, the probabilistic algorithms, such as Kalman Filter (KF, [1], [2]), Particle Filter (PF, [3], [4]) and include in the model this noisy behavior, generally by means of a probabilistic model.

In any case, the objective is to calculate the posterior probability ($p(\vec{x}_t|\vec{y}_{1:t})$) of the state vector $\vec{x}_t$, which informs about the position of the tracked objects, by means of the Bayes rule, and through a recursive two steps standard estimation process (prediction-correction), in which at least, some of the involved variables are stochastic.

The estimation process gets more complicated when a variable number of objects has to be tracked. In these situations an association algorithm is needed in order to run the estimator with the right input. The association problem is simplified if an only measurement for each object is in the output vector at each sample time. In contrast, the biggest the amount of information from each model is, the most reliable the estimation will be.

In the work presented here visual information is used in order to obtain as more position information from each tracked object as possible. Thus, the needed association algorithm has a high computational load but the reliability of the tracking process is increased.

In this context different alternatives have been tested by the research community, including Maximum Likelihood (ML), Nearest Neighbor (NN), Probabilistic Data Association (PDA [5], [6]), clustering, etc.

In this paper, 2 algorithms have been designed and tested in order to solve the multi-tracking application described:

- A KF with a PDA algorithm. As the state vector estimation in KF is represented by a unimodal density, the algorithm can only be used to track a single object. Therefore, the multi-tracking task based on a KF will have as many estimators as objects to track. In the proposed approach, a PDAF is used in order to decide which is the probability of each measurement to be considered in each object estimator. The joint of both algorithms, KF and PDAF, conform the KFPDAF algorithm.
- A PF used as a multi-modal estimator ([7], [8], [9], [10]). In this case, the multimodality of the filter is exploited to perform the estimation task for various models with a single PF. A clustering algorithm is used as association process in the estimation, whose deterministic behavior is also exploited in order to increase the multi-tracker robustness. The designed algorithm is called XPFCP.

Moreover, the conclusions extracted from the estimation results obtained with the proposed algorithms are presented.

The observation model used for the multi-tracking application presented is based on a stereo-vision system formed by two statically mounted on the robot that moves in

1-4244-0830-X/07/$20.00 ©2007 IEEE.

the environment. The vision system extracts pixels belonging to objects' edges in the pair of images captured at each sample time, and calculates their 3D position in the robot environment through the epipolar geometry that relates the cameras. Edge points in the image have been chosen to characterize the elements in the environment due to their robustness against light conditions and the simplicity of the vision algorithm used to extract and handle them.

The stereo-vision process described has been developed by the authors for this tracking application, and is explained in detail in [11].

II. Kalman Filter as a Position Estimator

Kalman Filter (KF) provides the optimal implementation of the Bayes estimator [2] when the system to which it is applied is linear, and the output and dynamics related noises are Gaussian with zero mean. This definition justifies the need of a more detailed analysis of the model to be used with the KF estimator.

A. *The Kalman Filter and the Position Model*

The application of the KF requires a model definition in the terms shown in the following expression:

$$\begin{matrix} \vec{x}_{t+1|t} = A\cdot\vec{x}_t + B\cdot\vec{u}_t + \vec{q}_t \rightarrow \vec{x}_{t+1|t} = f(\vec{x}_t,\vec{u}_t,\vec{q}_t) \\ \vec{y}_t = H\cdot\vec{x}_t + \vec{r}_t \rightarrow \vec{y}_t = h(\vec{x}_t,\vec{r}_t) \end{matrix} , \quad (1)$$

where the equations on the right are a general representation of the system behavior, and the ones on the left are used when the model is linear, accomplishing the first Kalman Filter restriction.

In the application of interest the state vector matches with the output one, so matrix H is equal to the entity one, therefore the observation model can be rewritten as follows:

$$\vec{y}_t = \vec{x}_t + \vec{r}_t \quad (2)$$

In (1) the input vector is obtained from $\vec{y}_t$ and $\vec{x}_{t|t-1}$:

$$\vec{u}_t = \frac{\vec{y}_t - \vec{x}_{t|t-1}}{T_s} \quad (3)$$

Applying the dynamics model definition in (1) to the estimation problem of interest:

$$\vec{x}_{t+1|t} = \begin{bmatrix} X_{t+1|t} \\ Z_{t+1|t} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Z_t \end{bmatrix} + \begin{bmatrix} T_s & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} v_{X,t} \\ v_{Z.t} \end{bmatrix} + \vec{q}_t , \quad (4)$$

In (4), $[X \quad Z]'$ are the Cartesian coordinates of the object centroide projection in the moving plane.

In (1), the noise model has also been included as $\vec{q}_t$ and $\vec{r}_t$. Some other specifications of the KF are also related with these noise models:

- Both the state noise and the measurements noise have to be Gaussian. This specification is sometimes difficult to assert, and in that case the optimal estimation solution guaranteed by the KF will not be completed, and convergence problems in the estimation loop may come out. This kind of problem has not appeared in the estimation process developed in this work.
- The covariance matrixes Q and R that respectively characterize the noise vectors $\vec{q}_t$ and $\vec{r}_t$ in the Gaussian noise models, should be diagonal in order to achieve the KF specifications. This assessment cannot be easily assured in most of cases. To ensure the achievement of this specification, an empirical study has been developed in order to obtain these two matrixes, for the application of interest, whose result shows that the diagonal approximation can be applied.

The KF functionality is shown in Fig. 1, where the two main steps of the probabilistic estimator can be seen. A detailed description of them can be found in [2].

Applying a KF to the multi-tracking problem of interest imposes some other problems. As the object dynamics changes from one object to other, a single KF can only be used to track one object. It is also possible to use an only estimator for all the objects if the state vector $\vec{x}_t$ size is dynamically adapted to include the state variables of objects as they appear or disappear in the scene. Nevertheless, this option has shown to be computationally expensive, and thus, the real-time execution is difficult to achieve with it ([12], [13], [14]).

Thus, most solutions to the same multi-estimation problem that are based on a KF are developed using one KF for each object to track (see [1], [15] and [16]).

In any case, as explained in the introduction paragraph, an association algorithm is needed in order to develop correctly the correction step of the algorithm.

B. *The PDAF and its Insertion in the Kalman Filter*

The Probabilistic Data Association Filter (PDAF) is a Bayesian approach to solve the data association problem exposed. In the association process, the influence or probability of every measurements to every object ($\beta_{i,j}$) is calculated.

Thus, each $\beta_{i,j}$ is the probability of *ith* measurement being originated by the object *jth* ([6]).

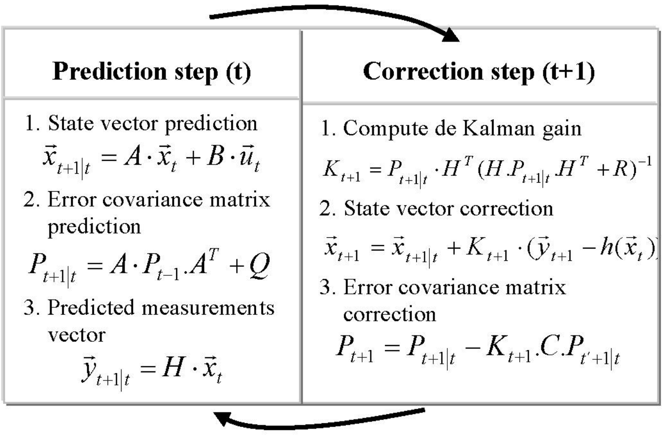
Prediction step (t)
1. State vector prediction
$\vec{x}_{t+1|t} = A\cdot\vec{x}_t + B\cdot\vec{u}_t$
2. Error covariance matrix prediction
$P_{t+1|t} = A\cdot P_{t-1}.A^T + Q$
3. Predicted measurements vector
$\vec{y}_{t+1|t} = H\cdot\vec{x}_t$
Correction step (t+1)
1. Compute de Kalman gain
$K_{t+1} = P_{t+1|t}\cdot H^T(H.P_{t+1|t}.H^T + R)^{-1}$
2. State vector correction
$\vec{x}_{t+1} = \vec{x}_{t+1|t} + K_{t+1}\cdot(\vec{y}_{t+1} - h(\vec{x}_t))$
3. Error covariance matrix correction
$P_{t+1} = P_{t+1|t} - K_{t+1}.C.P_{t'+1|t}$

Fig. 1. Description of the KF functionality

To obtain an assignation result, only those measurements whose influence is above a threshold (also called gate) are related to each object, in the estimation process.

This influence of each measurement in every object can be calculated according to different parameters. In the work presented, the characteristic chosen is the Euclidean distance from the *ith* measurement to the *jth* object predicted output vector ($d_{i,j}$), and a normal distribution is used in order to obtain all probability values, as follows:

$$\beta_{i,j} = e^{-\frac{1}{2}\cdot\frac{\min(d_{i,j}^2)}{dMax^2}} \;/\; i = 1:m, j = 1:k, d = \left|\vec{y}_i - \vec{y}_{t|t-1}\right|, \quad (5)$$

where $dMax = \sqrt{2}\cdot\sigma$ represents the effect of the validation gate in the probability computation; m is the number of measurements extracted from the environment at each iteration and k is the number of objects being tracked.

The predicted output vector ($\vec{y}_{t|t-1}^{(i)}$) is calculated for each object as follows:

$$\vec{y}_{t|t-1}^{(i)} = h(\vec{x}_{t|t-1}^{(i)}) = \vec{x}_{t|t-1}^{(i)}, \quad (6)$$

due to the model specifications commented in previous section.

It has to be notice that $\vec{x}_{t|t-1}^{(i)}$ is the deterministic estimated value of the *ith* object position in the environment.

If the distance $d_{i,j}$ from the *ith* measurement to the closest object is larger than the validation gate, the measurement is supposed to b related with a new object, and a new KF is created in order to track it. This way, the tracking algorithm adapts itself on-line to estimate the position of a variable number of objects.

When applying a PDAF to the KF estimation, the equation used to obtain the state vector corrected value in the correction step is modified (see Fig. 1), changing the standard innovation parameter ($r_t = K_{t+1}\cdot(\vec{y}_{t+1} - h(\vec{x}_t))$) by the combined one:

$$r_{j,t} = \sum_{i=1}^{mj} \beta_{i,j}\cdot\left(\vec{y}_t^{(i)} - h(\vec{x}_{t|t-1}^{(i)})\right) \quad (7)$$

The combined innovation is computed over the mj measurements assigned to the *jth* track at each iteration.

The final structure of the KFPDAF estimation algorithm is shown in Fig. 2.

III. Particle Filter as a Position Estimator

A Particle Filter (PF) is used as a multimodal estimator to obtain the position and speed of multiple objects of the environment.

The PF is a Bayes filter particularization where the density related to the posterior estimation (called belief) is discretized.

A detailed description of the PF mathematical base can be found in [3] and in [4].

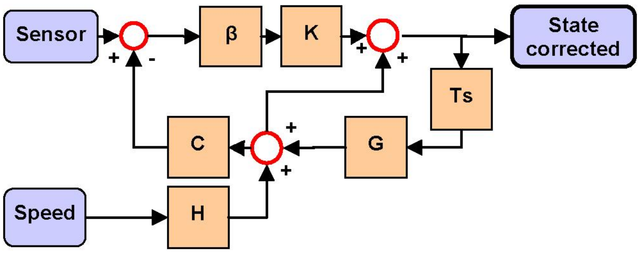

Fig. 2. Block diagram showing the functionality of the multi-tracker designed, including the PDAF in the KF loop

As the state vector is not discretized, the PF is more accurate in its estimation than the KF or other implementations of the Bayes filter, like the Monte-Carlo ones.

Moreover, due to the same reason, the computational load of this Bayes filter form is lower than others', and thus the PF is more adequate to implement real time estimation.

Finally, PFs presents an interesting characteristic for multi-tracking applications: the possibility of representing multiple estimation hypotheses with a single algorithm, through the multimodality of the belief. This facility is not available if using the KF.

For all these reasons, the PF has been thought as the most appropriated algorithm to develop a multi-tracking system.

A. The Particle Filter

Most of the solutions of the tracking problem, based on a PF, do not exploit the multimodal character of the algorithm in order to implement the position estimation of multiple objects task. The main reason of this fact is that, in general, the association process needed to accomplish the multi-estimation task is very expensive in execution time (as it is the case in the solution presented in [9]) or lacks of robustness (as it is the case in [17]).

The XPFCP presented here is a multimodal estimator based on a single PF that achieves the real-time and robustness specification, thanks to a clustering process that is used as association process in the estimation loop. The functionality of the XPFCP is presented in the following paragraphs.

The main loop of a standard Bootstrap PF ([3], [4]) starts at time t with a set $S_{t-1} = \left\{\vec{x}_{t-1}^{(i)}, \widetilde{w}_{t-1}^{(i)}\right\}_{i=1}^{n}$ of n weighted random samples or particles, representing the state vector belief $p(\vec{x}_{t-1}|\vec{y}_{1:t-1})$, estimated at the previous step $t-1$. The rest of the process is developed in three steps, as follows:

1. *Prediction step:* The particles are propagated by the motion model $p(\vec{x}_t|\vec{x}_{t-1})$ to obtain a new set $S_{t|t-1} = \left\{\vec{x}_{t|t-1}^{(i)}, \widetilde{w}_{t-1}^{(i)}\right\}_{i=1}^{n}$ that represents the prior distribution of the state vector at time *t*, $p(\vec{x}_t|\vec{y}_{1:t-1})$.
2. *Correction step:* The weight of each particle $\vec{w}_t = \left\{w_t^{(i)}\right\}_{i=1}^{n} \equiv w(\vec{x}_{0:t})$ is then obtained comparing the output vector $\vec{y}_t$ and its predicted value based on the prior

estimation $h(\vec{x}_{t|t-1})$. In the Bootstrap version of the filter, these weights are obtained directly from the likelihood function $p(\vec{y}_t|\vec{x}_t)$, as follows:

$$w(\vec{x}_{0:t}) = w(\vec{x}_{0:t-1}) \cdot \frac{p(\vec{y}_t \mid \vec{x}_t) \cdot p(\vec{x}_t \mid \vec{x}_{t-1})}{q(\vec{x}_t | \vec{x}_{0:t-1}, \vec{y}_{1:t})} \qquad (8)$$
$$\xrightarrow[q(\vec{x}_t|\vec{x}_{0:t-1},\vec{y}_{1:t}) \propto p(\vec{x}_t|\vec{x}_{t-1})]{} w(\vec{x}_{0:t}) = w(\vec{x}_{0:t-1}) \cdot p(\vec{y}_t \mid \vec{x}_t)$$

3. *Selection step:* Using the weights vector $\vec{w}_t = \{w_t^{(i)}\}_{i=1}^{n}$, and applying a resampling scheme ([3], [4]), a new set $S_t = \{\vec{x}_t^{(i)}, \widetilde{w}_t^{(i)}\}_{i=1}^{n}$ is obtained with the most probable particles, which will represent the new belief $p(\vec{x}_t|\vec{y}_{1:t})$.

In the application of interest, the standard PF can be used to robustly estimate the position of any kind of a single object defined through its motion model, but it cannot be directly used to estimate the position of appearing objects because there is not a process to relate particles to the new estimations.

In order to adapt the standard PF to be used in a multi-tracking application, some modifications must be included in the basic algorithm. In [17] an adaptation of the standard PF for the multi-tracking task is proposed. Nevertheless, the algorithm described there is not applied to a real tracking application because it is not robust enough.

The solution that is used in this paper was proposed by the authors in [18], and is called XPFCP. The algorithm is an extension of the PF that includes a clustering algorithm to improve the behavior of the extended PF in [17].

The clustering algorithm, whose functionality is presented in next section, organizes the set of measurements in clusters that represent all objects in the scene. These clusters are then wisely used in the multimodal estimator. Two innovations are included in the standard PF to achieve the multimodal behavior:

- *A new re-initialization step:* In this step, previous to the prediction one, $n_{m,t}$ from the n total number of particles that represent the belief $p(\vec{x}_{t-1}|\vec{y}_{1:t-1})$ at $t-1$ are directly substituted by measurements from the observation vector at that execution time, $\vec{y}_{t-1}$. With this modification, measurements related to appearing objects in the scene have a representation in the priori distribution $p(\vec{x}_{t-1}|\vec{y}_{1:t-1})$. To improve the robustness of the estimator, the new particles are not selected randomly from the array of measurements $\vec{y}_{t-1}$ but from the k clusters $G_{1:k,t-1}$. Choosing measurements from every cluster ensures a probable representation of all elements to track, and therefore, an increased robustness of the multi-tracker. Thanks to this re-initialization step the belief dynamically adapts itself to represent the position hypothesis of the different objects in the scene.
- *At the Correction step:* This step is also modified from the one in the standard PF. On one hand, only $n-n_{m,t}$ samples of the particle set have to be extracted in this step, as the $n_{m,t}$ resting ones would be inserted with the re-initialization one. On the other hand, the clustering process is also used in this step, because the importance sampling function $p_i(\vec{y}_t|\vec{x}_t^{(i)})$ used to calculate each particle weight $w_t^{(i)}$ is obtained from the similarity between the particle and the k cluster centroides $g_{1:k,t}$. Using the cluster centroides to weight the particles related to the newly appeared objects, the probability of these particles is increased, improving the robustness of the new estimation hypotheses. Without the clustering process, the solution proposed in [17] rejects these hypotheses, and thus, the multimodality of the PF cannot be robustly exploited.

Fig. 3 shows the XFPCP functionality, described in previous paragraphs.

B. The clustering

Two different clustering algorithms have been developed for using in the XPFCP: an adapted version of the K-Means for a variable number of clusters; and a modified version of the Subtractive fuzzy clustering.

Their reliability is similar, but the proposal based on the standard K-Means shows higher robustness rejecting outliers in the measurements set. A more detailed analysis of these algorithms can be found in [19].

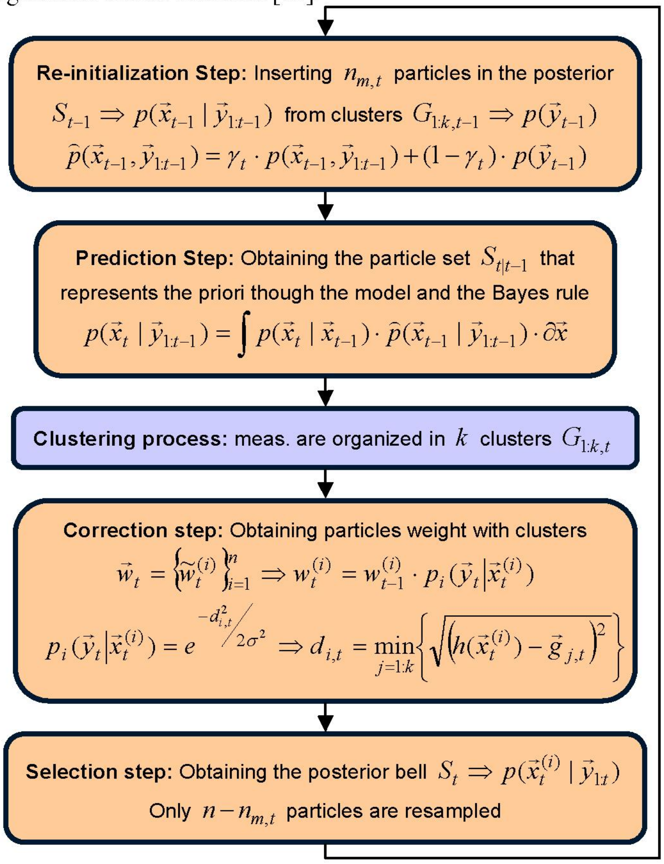
Re-initialization Step: Inserting $n_{m,t}$ particles in the posterior
$S_{t-1} \Rightarrow p(\vec{x}_{t-1} \mid \vec{y}_{1:t-1})$ from clusters $G_{1:k,t-1} \Rightarrow p(\vec{y}_{t-1})$
$\widehat{p}(\vec{x}_{t-1}, \vec{y}_{1:t-1}) = \gamma_t \cdot p(\vec{x}_{t-1}, \vec{y}_{1:t-1}) + (1-\gamma_t) \cdot p(\vec{y}_{t-1})$
Prediction Step: Obtaining the particle set $S_{t|t-1}$ that represents the priori though the model and the Bayes rule
$p(\vec{x}_t \mid \vec{y}_{1:t-1}) = \int p(\vec{x}_t \mid \vec{x}_{t-1}) \cdot \widehat{p}(\vec{x}_{t-1} \mid \vec{y}_{1:t-1}) \cdot \partial\vec{x}$
Clustering process: meas. are organized in k clusters $G_{1:k,t}$
Correction step: Obtaining particles weight with clusters
$\vec{w}_t = \{\widetilde{w}_t^{(i)}\}_{i=1}^{n} \Rightarrow w_t^{(i)} = w_{t-1}^{(i)} \cdot p_i(\vec{y}_t|\vec{x}_t^{(i)})$
$p_i(\vec{y}_t|\vec{x}_t^{(i)}) = e^{-d_{i,t}^2 / 2\sigma^2} \Rightarrow d_{i,t} = \min_{j=1:k}\left\{\sqrt{\left(h(\vec{x}_t^{(i)}) - \vec{g}_{j,t}\right)^2}\right\}$
Selection step: Obtaining the posterior bell $S_t \Rightarrow p(\vec{x}_t^{(i)} \mid \vec{y}_{1:t})$
Only $n-n_{m,t}$ particles are resampled

Fig. 3. XPFCP Flowchart

Two main modifications to the standard K-Means functionality have been included in the proposal:

1. It has been adapted in order to handle a variable and initially unknown number k of clusters $G_{1:k}$, by defining a threshold $distM$ in the distance $d_{i,1:k}$ used in the clustering process.
2. A cluster centroides' prediction process is included at the beginning of the algorithm in order to minimize its execution time. Whit this information, the process starts looking for centroides near their predicted values $\vec{g}_{0,1:k,t} = \vec{g}_{1:k,t|t-1}$.

A validation process is also added to the clustering in order to increase the robustness of the algorithm to spurious measurements. This process is useful when outliers produce a cluster creation or deletion erroneously. The validation algorithm functionality is the following:

- When a new cluster is created, it is converted into a candidate that will not be used in the XPFCP until it is possible to follow its dynamics.
- The same procedure is used to erase a cluster when it is not confirmed with new measurements for a specific number of times.

The validation process is based in two parameters, which are calculated for each cluster:

- *Distance between the estimated and the resulting clusters centroide.* The centroides estimation process, already commented, is also used in the validation process. The estimated value of the centroides $\vec{g}_{1:k,t|t-1}$ is compared with its final value at the end of the clustering process $\vec{g}_{1:k,t}$, in order to obtain a confidence value for the corresponding cluster validation.
- *Cluster likelihood.* A cluster probability value is calculated as a function of each cluster number of members, $L_{1:k}$.

Some application results of this multimodal estimator proposed by the authors to the multi-tracking task are shown in the following section. The effectiveness of the clustering proposal in the tracking application is demonstrated there.

IV. Results

The algorithm has been run on a real-time system, mounted on board of a commercial robotic platform (Pioneer 2AT).

The sensory system (a pair of B&W digital and synchronized cameras) has been statically mounted and calibrated on top of the robot, at a height of 1.7m.

The stereo-vision runs in the same processing platform, generating position information of the obstacles in the environment at a 15fps rate.

In this situation, different tests have been developed showing the functionality of the designed trackers in various environmental conditions.

One up to six static and dynamic objects have been tracked with the proposed algorithms in videos of almost 2 minutes length (1653 frames). Different types of dynamic (persons and a robot) and static objects (a small trash, a tall box, etc.) are correctly tracked in almost all situations.

The following conclusions can be extracted from these experiments.

- A missing error rate of less than a 10% is achieved by both algorithms in complex situations (7.5% for the XPFCP and 8.4% for the KFPDAF). These errors are due to occlusions most of times. In these situations, the set of measurements related to the occluded object is very poor, and XPFCP manages better these situations than KFPDAF.
- Generating 2 or more hypotheses for a single object is another type of error found in the results. This event is called duplication error, and in the global set of experiments it has been generated a rate of 4.6% if running KFPDAF and only a 1.3% if running XPFCP. The main reason for this error is the sensibility of the PDAF to the value of the gating parameter.
- Simpler experiments (tracking an only person and a static object) give a result free of errors with both proposals.

In simplest experiments, the mean execution time of XPFCP is slightly bigger than the one of KFPDAF, but in complex situations where more than 4 or 5 objects are being tracked, the execution time of the XPFCP is almost half of the KFPDAF one. This fact is due to the already commented high computational load of the PDAF, which becomes important when the amount of measurements is high.

In general it can be concluded, that the PF proposal gives better results in complex situations like the ones proposed than the KF one. Fig. 4 shows some of the results generated by the designed algorithms in a complex situation, in which 4 objects are being tracked.

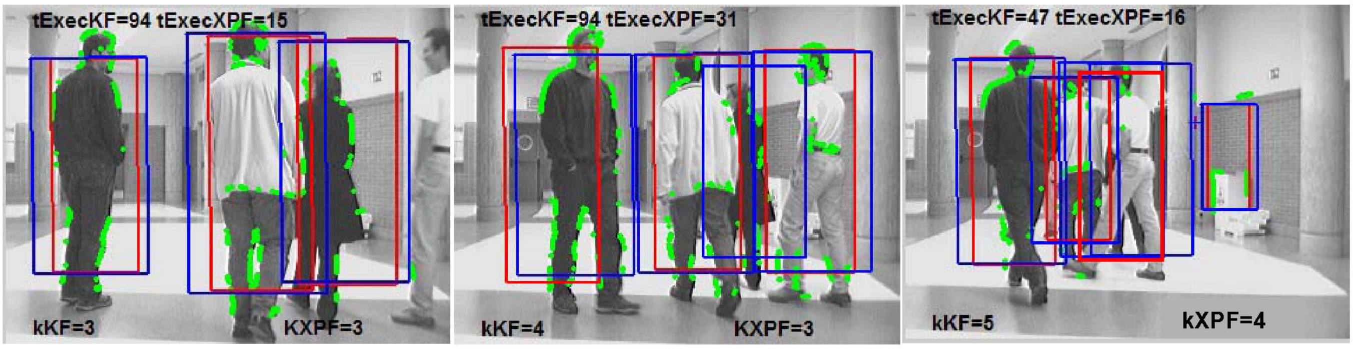

Fig. 4. Images of the tracking results obtained with KFPDAF algorithm (in blue rectangles) and the XPFCP (in red rectangles) in a complex situation. A standard PC with an Intel Dual Core processor at a 2.4GHz with 2Gbytes of RAM has been used to run the algorithms in these experiments.

A colored rectangle has been drawn surrounding each tracked object. Red ones display the results obtained with the PF proposal, and blue ones the results generated by the KF one. Position measurements extracted with the stereo-vision system are drawn in green in the images.

In the figure, the number of objects and the execution time generated by each tracker is also displayed.

The images in the figure are sequentially presented from left to right. Besides, they have been selected in order to show some of the typical results generated by the trackers in a complex situation:

- The frame in the left shows a correct result of both algorithms. Three persons in the middle of the scene are properly tracked, and the one in the left cannot be tracked as no position measurement related to it has been obtained.
- The frame in the middle shows a situation in which the XPFCP generates a missing error with the occluded person. Both algorithms present some reliability problems in this kind of situations as mentioned previously.
- The frame in the right displays a situation in which the KFPDAF generates a duplication error with the person on the right. As explained previously, this error is due to the algorithm sensibility to the gating parameter $dMax$.

In the image, it can also be appreciated that the KFPDAF execution time is bigger than the XPFPC one. This fact is due to the association algorithm, as explained previously.

V. Conclusions

In this paper, two proposals for a robotic multi-tracking application have been developed based on two Bayesian solutions: a KF and a PF. The probabilistic estimators have been modified, including an association algorithm to each of the solutions in order to achieve the functionality pursuit, obtaining the KFPDAF and the XPFCP. Both algorithms have been described, and their reliability has been demonstrated, in different environmental situations.

The XPFCP seems to be more reliable and robust than the KFPDAF, mainly due to the last one sensibility to the gating parameter included in the PDAF. This is the reason for the authors to conclude that a PF may be more adequate for multi-tracking tasks in complex situations, and a KF should be the chosen solution in simpler ones, such as surveillance applications in low populated areas.

Acknowledgment

This work was supported jointly by the Ministry of Science and Technology under RESELAI project (reference TIN2006-14896-C02-01) and Regional Government of Madrid and the University of Alcala under VISO project (reference CCG06-UAH/DPI0745).

References

[1] D.B. Reid. “An algorithm for tracking multiple targets”, *IEEE Trans. on Automatic Control,* vol. 24, no. 6, pp. 843-854, December 1979.

[2] G. Welch, G. Bishop, Eds. *An Introduction to the Kalman Filter,* 2001.

[3] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp. “A tutorial on particle filters for online nonlinear non-Gaussian Bayesian tracking”, *IEEE Trans. on Signal Processing,* Vol. 50, nº 2, pp: 174-188, February 2002.

[4] N.J. Gordon, D.J Salmond, A.F.M. Smith. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”, *IEE Proc. Part F,* Vol. 140, nº 2, pp: 107-113, April 1993.

[5] Y. Bar-Shalom, T. Fortmann. *Tracking and Data Association (Mathematics in Science and Engineering, V.182),* Academic Press, ISBN: 0120797607, January 1988.

[6] C. Rasmussen, G.D. Hager. “Probabilistic data association methods for tracking multiple compound visual objects”, *IEEE Trans. on Pattern Analysis andMachine Intelligence,* vol. 23, nº. 6, pp. 560-576, 2001.

[7] D. Schulz, W. Burgard, D. Fox, A. B. Cremers. “People tracking with mobile robots using sample-based joint probabilistic data association filters”, *International Journal of Robotics Research,* Vol. 22, nº 2, pp: 99-116, February 2003.

[8] Z. Khan, T. Balch, F. Dellaert. "MCMC-Based particle filtering for tracking a variable number of interacting targets", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 27, nº 11, pp. 1805-1918, November 2005.

[9] A. Almeida, J. Almeida, R. Araujo. "Real-time tracking of moving objects using particle filters", *in Proc. of the IEEE International Symposium on Industrial Electronics (ISIE05),* ISBN: 0-7803-8738-4. Vol. I-IV, pp. 1327-1332, Dubrovnik, June 2005.

[10] D. Tweed, A. Calway. “Tracking many objects using subordinated condensation”, *in Proc. of the British Machine Vision Conference (BMVC02),* ISBN: 1901725197, pp. 283-292, October 2002.

[11] M. Marron, M.A. Sotelo, J.C. Garcia, D. Fernandez, I. Parra. “3D-visual detection of multiple objects and environmental structure in complex and dynamic indoor environments”, *in Proc. of the 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON06),* ISBN: 1-4244-0136-4, pp. 3373-3378, Paris, 2006.

[12] C. Hue, J.P. Le Cadre, P. Pérez. “A particle filter to track multiple objects”, *IEEE Trans. on Aerospace and Electronic Systems,* vol. 38, nº. 3, pp. 791-812, July 2002.

[13] J. MacCormick, A, Blake. “A probabilistic exclusion principle for tracking multiple objects”, *in Proc. of the Seventh IEEE International Conference on Computer Vision (ICCV99),* vol. 1, pp. 572-578, Corfu, September 1999.

[14] K. Smith, D. Gatica-Perez, J.M. Odobez. “Using particles to track varying numbers of interacting people”, *in Proc. of the Fourth IEEE Conference on Computer Vision and Pattern Recognition (CVPR05),* pp. 962-969, San Diego, June 2005.

[15] T. Schmitt, M. Beetz, R. Hanek, S. Buck. “Watch their moves applying probabilistic multiple object tracking to autonomous robot soccer”, *in Proc. of the Eighteenth National Conference on Artificial Intelligence (AAAI02),* ISBN: 0-262-51129-0, pp. 599-604, Edmonton, July 2002.

[16] S. Coraluppi, C. Carthel. "Multi-hypothesis sonar tracking", *in Proc. of the Seventh International Conference on Information Fusion (FUSION04),* ISBN: 91-7056-115-X, Vol. 1, pp. 33-40, Stockholm, June 2004.

[17] E.B. Koller-Meier, F. Ade. “Tracking multiple objects using a condensation algorithm”, *Journal of Robotics and Autonomous Systems,* Vol. 34, pp. 93-105, February 2001.

[18] M. Marrón, J.C. García, M.A. Sotelo, D. Fernandez, D. Pizarro. “XPFCP: An extended particle filter for tracking multiple and dynamic objects in complex environments", *in Proc. of the IEEE International Symposium on Industrial Electronics 2005 (ISIE05),* ISBN: 0-7803-8738-4. vol. I-IV, pp. 1587-1593, Dubrovnik, June 2005.

[19] M. Marrón, M.A. Sotelo, J.C. García, J. Broddfelt. “Comparing improved versions of ‘K-Means’ and ‘Subtractive’ clustering in a tracking applications”, *in Proc. of the Eleventh International Workshop on Computer Aided Systems Theory (EUROCAST07),* ISBN: 978-84-690-3603-7, pp. 252-255, Las Palmas de Gran Canaria, February 2007.