“XPFCP”: An Extended Particle Filter for Tracking Multiple and Dynamic Objects in Complex Environments

M. Marrón[*], M. A. Sotelo[*], J.C. García[*], D. Fernández[*], D. Pizarro[*]

[*]Electronics Department, University of Alcala, Alcalá de Henares, SPAIN

Abstract—**The work presented in this paper explores a new solution for tracking multiple and dynamic objects in complex environments. An extended particle filter (XPF) is used to implement a multimodal distribution that will represent the most probable estimation for each object position. A standard particle filter (PF) cannot be used with a variable number of obstacles, and some other solutions have been tested in different previous works, but most of them are very expensive in time and memory resources at least for a high number of obstacles to be tracked. The solution exposed here includes a clustering procedure that increases the robustness of the probabilistic process to adapt itself on-line to the variable number of clusters. The presented algorithm has been tested with sonar and stereovision measurements and some results included in the paper show the efficiency of the proposed work.**

I. Introduction[a]

Probabilistic algorithms have shown their reliability solving estimation problems since more than fifty years ago. These methods were quickly applied to solve positioning problems in robot navigation ([1]).

The particle filter (PF) was developed in the 90s as the way to implement a discrete distribution and to develop a sampling weight representation of the Bayesian filter, in a computational cheap fashion with the works of Isard and Blake some years after ([3][4]).

The problem of multiple objects tracking appeared soon with the first autonomous navigators, and soon the probabilistic algorithms were applied to solve the problem ([5][6]).

In this point JPDAF filters have shown the highest reliability ([7]). These techniques use a standard PF or a Kalman Filter to track each object which is not very efficient to work with a dynamic number of objects, and include an association process to organize the measurements among the different filters. The algorithm has a high computational cost, which on the other hand, depends on the number of tracked objects.

The challenge of the XPFCP (XPF with pre-clustering) presented here is to use a single multimodal distribution to model the different obstacles in the environment, what will do the global execution time constant and lower than with the techniques mentioned before.

The work exposed here was firstly presented in [8], and is based on the extended particle filter (XPF) for multiple obstacles from [9].

The algorithm presented in [8] as a solution to the multiple tracker is based on a adapted standard PF to dynamically include particles from the new detected objects, and includes a pre-clustering to improve the robustness of the final estimator. In that paper the functionality of the XPFCP was presented and some results of the work applied to sonar measurements were also shown.

In this paper, some interesting considerations improved in the algorithm are presented, and the resulting XPFCP is applied not only to sonar measurements but also to stereovision data, with real results shown at the end.

II. The Model

The main objective of the XPFCP presented is to model the movement of the objects that surround the robot in a complex environment.

To achieve this aim the different objects detected by the sensors around the robot have been characterized by a dynamics model, in which the state vector includes the position speed and acceleration in Cartesian coordinates, relative to the robot ones.

The following equations show the proposed system:

$$\vec{a}_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \\ \ddot{x}_t \\ \ddot{y}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 & T^2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & f_d() & 0 \\ 0 & 0 & 0 & 0 & 0 & f_d() \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \\ \ddot{x}_{t-1} \\ \ddot{y}_{t-1} \end{bmatrix} + \vec{r}_t \text{, state eq. (1)}$$

$$\vec{c}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \\ \ddot{x}_t \\ \ddot{y}_t \end{bmatrix} + \vec{o}_t \text{, measurem. eq. (2)}$$

where:

- $f_d()$ is the non-linear part of the model, and as it can be seen, is used to modify the acceleration of the object. This function depends on different parameters of the instantaneous configuration of the detected environment and the robot movement.

[a] This work has been financed by the by the Spanish administration through Research Project CICYT DPI2002-04064-05-04

- $\vec{r}_t$ and $\vec{o}_t$ are the noise vectors respectively related to the state vector (the system model itself) and the output vector (measurements).

The measurements noise vector has to be characterized for each one of the sensors used in the estimator, in an off-line previous process. In this work these random variables have been identified as Gaussian and white under the specification of a previous calibration of the sensor.

The system noise vector is empirically characterized also as Gaussian and white, taking into account that the second order model used for the objects movement is accurate enough.

In any case the value of these two vectors is very important in the estimator evolution, as $\vec{r}_t$ will state the spread capability of the particles that identify an object, and $\vec{o}_t$ is going to be used in the calculus of the particle set likelihood. Some empirical work was needed to adapt these parameters to sensor configuration.

A. The dynamic model to be tracked with sonar data

A sonar ring all around the robot was used to observe the environment with the ultrasonic sensors. These sensors have a very poor resolution and range.

In the tests made with this sonar, 16 sensors were used with a range from 3cm to 3m and a millimeter resolution.

The main problem of the sonar ring is the need of a triggering synchronization topology, which results in a high sample time of each of the sensors. This value was of about 300ms in the robotic platform used in the experiments done, and a standard value in most of the typical robotic commercial platforms[b].

On the other hand, the mentioned noise identification was done for these sensors in the platform used for the tests, obtaining that in this case $\vec{o}_t$ was characterized by a Gaussian distribution with $\mu = 0$ (white noise) and $\sigma = 4$ mm.

With these specifications, it is easy to guess that the ultrasonic sensor is going to give poor information to obtain reliable knowledge about the dynamic environment around the robot. The main problem will come with the low sampling rate of the measurements, what will cause a late detection of fast obstacles that appear near the robot.

Then again, the low cost and low pre-processing need of sonar data make their use very striking. That is the reason for using ultrasonic sensors in the tests shown.

B. The dynamic model to be tracked with vision data

A stereovision system has been used to obtain vision data of the objects position around the robot. The stereovision process is based on the epipolar geometry between the two cameras ([10]).

At the same time, only moving points are extracted from the corresponding data set, obtained with the epipolar geometry at each pair of images.

In this case, the measurement vector is increased in a third dimension, as shown in (3), and in the same line, the state vector equation could be also increased with this third coordinate (z):

$$\vec{c}_t = \begin{bmatrix} x_t & y_t & z_t \end{bmatrix}' \qquad (3)$$

Nevertheless, the third dimension is only used, in this work, to filter the possible noise from the corresponding data set at each pair of images. Some improvements in the clustering, which are currently in progress, use the third dimension to increase the robustness of the segmentation.

The resolution obtained with the stereovision system is similar to the one of the sonar, but it has a bigger range (until 20m). This fact is important, as it allows knowing the obstacles position from a higher distance, and using this data to implement obstacle avoidance behaviors with higher reliability.

The sampling rate of the stereovision sequences is about 50ms (20fps). This rate is also better than the one obtained with the ultrasonic sensors, and make possible a faster reaction against obstacles near to the robot.

The stereovision system also needs a calibration procedure, which includes an identification of the noise attached to the measurements that it provides. The empirical tests give a vector $\vec{o}_t$ characterized by a Gaussian distribution with $\mu = 0$ (white noise) and $\sigma = 50$ mm.

Though stereovision data are noisier, the higher sampling rate of this sensory system allows very reliable results as shown afterwards in the paper.

III. The Extended Particle Filter

A. The XFP proposed

The main loop of a standard PF ([11][12][13]) starts at time *t* with a set $S = \{s_i / i = 1..N\}$ of random particles representing the posterior distribution of the state vector to be estimated $p(\vec{a}_{t-1} | \vec{c}_{1:t-1})$ at the previous time step.

These particles are propagated by the system dynamics to obtain a new set S' that represents the prior distribution of the state vector at time *t*, $p(\vec{a}_t | \vec{c}_{1:t-1})$.

The weight of each particle $W = \{w_i / i = 1..N\}$ is then obtained based on the comparison of the measured output vector and the predicted one based on the prior estimations.

Applying the selected resample scheme, a new set S'' is obtained with the most probable particles that will be the new $p(\vec{a}_t | \vec{c}_{1:t})$.

The standard PF estimates quite well the evolution of any kind of a single object defined by its model, but as mentioned in [9], two constraints disable its performance tracking multiple and dynamic objects:

- First of all, the standard PF is not able to estimate new appearing objects in the environment as all measurements related with new objects would be rejected at the resampling stage as no particles would be similar to these measurements. To solve this problem, a re-initialization stage is introduced in the standard loop of a PF to insert new particles directly from the measurements in the sample set. On the other hand, the resampling stage will also need to be

[b] Higher sampling rate is possible with other synchronization topologies in the sonar ring.

modified to allow the incorporation of the new particles in the sample set.

- Secondly, to track multiple objects, the importance sampling stage has also to be modified as the likelihood has to be calculated depending on the similarity between each particle and the measurements from its corresponding object. If it is not the case, the particles related to the object can also be rejected as mentioned in the previous constraint. To solve this other problem, a modification has to be inserted in the function to calculate the particles' weights according to their corresponding measurements.

Fig. 1 presents a graphical description of the XPF explained before. The functionality of the modified re-initialization and importance sampling stages are detailed in the following paragraphs.

B. The XPF tracking multiple and dynamic objects

1) The re-initialization stage: A re-initialization of the sample set S at each time step has to be done, inserting on it *M* samples directly from the output vector.

With this modification, the distribution $p_m(\vec{a}_{t-1}|\vec{c}_{t-1})$ from the new environmental configuration is combined with the posterior distribution $p_p(\vec{a}_{t-1}|\vec{c}_{1:t-1})$ to obtain a new expression for it:

$$p(\vec{a}_{t-1}|\vec{c}_{1:t-1}) = \gamma \cdot p_m(\vec{a}_{t-1}|\vec{c}_{t-1}) + (1-\gamma)\cdot p_p(\vec{a}_{t-1}|\vec{c}_{1:t-1}), \quad (4)$$

where γ is the factor that weights the distribution association up, and is fixed by the relation between the *M* samples inserted directly from the output vector measured at *t-1* and the total number of samples (*N*) in the particle set S ($\gamma = {}^{M}\!/_{N}$).

The single probability distribution will adapt itself over time to finally represent simultaneously the state vector of all different objects that exist in the scene at each step.

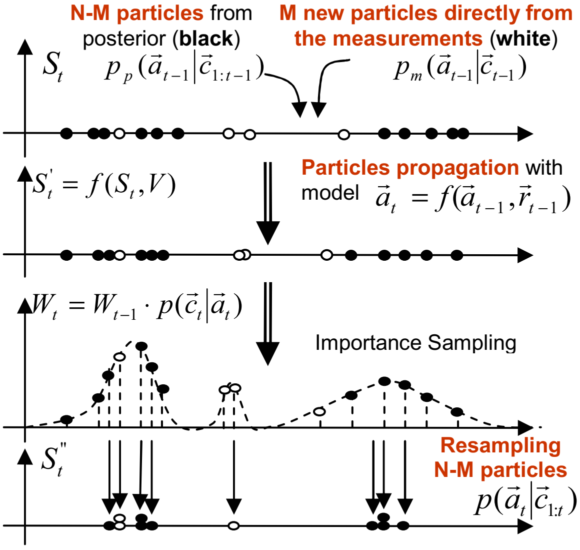

N-M particles from posterior (black)
M new particles directly from the measurements (white)
S_t
$p_p(\vec{a}_{t-1}|\vec{c}_{1:t-1})$
$p_m(\vec{a}_{t-1}|\vec{c}_{t-1})$
$S_t^{'} = f(S_t, V)$
Particles propagation with model $\vec{a}_t = f(\vec{a}_{t-1}, \vec{r}_{t-1})$
$W_t = W_{t-1}\cdot p(\vec{c}_t|\vec{a}_t)$
Importance Sampling
$S_t^{''}$
Resampling N-M particles
$p(\vec{a}_t|\vec{c}_{1:t})$

Fig. 1. Description of the extended PF functionality.

On the other hand, if measurements are randomly selected from all the input set the reinforcement of the new obstacles tracking is not ensured. This problem will be solved with an oriented selection of the *M* particles to directly insert at the re-initialization phase. The organization of the measurements to achieve this aim is obtained thanks to the proposed pre-clustering algorithm, presented in following paragraphs.

2) At the importance sampling stage: To enable the possibility of representing multiple objects in an unique distribution, the particle weights at the importance sampling have to be obtained according to the concrete object that each particle is tracking.

To achieve this objective, the base function of the likelihood calculus is modified as follows:

$$p(\vec{c}_t|\vec{a}_t) = e^{-u^2/2\sigma^2}, \text{ if } u = \min_m\left\{\sqrt{\left(\vec{c}_t^{\,i} - \vec{c}_t^{\,m}\right)^2}\right\}, \quad (5)$$

where $\vec{c}_t$ comes from the model show in (2), and $\vec{c}_t^{\,m}$ represents to direct measurements.

As (5) shows, the new distribution is obtained from the minimum distance from the particle to all measurements at each time step.

This function presents the only problem (as commented in [9]) of giving more weight to the objects that are measured with higher accuracy, and consequently of probably rejecting the particles related to the most poorly sensed objects. The pre-clustering included in the final XPFCP will also solve this problem, as it partially filters the measurements inaccuracy.

3) At the resample stage: To insert the new *M* particles as mentioned, the resample stage is also modified. In this case, only *N-M* samples have to be selected from the *N* existing at the S' sample set.

The resampling process, as well as the rest of the PF is for the rest equal to the standard PF. Again in this step the proposed pre-clustering can be used to choose selectively, the particles to be resampled, according to their clustering association.

IV. The Clustering

A. The modified k-means algorithm

As it has previously mentioned, a segmentation process has been designed to organize the measurements that come from the sensor in a variable number of clusters.

This process is based on a standard *kmeans* ([14]) with unknown initial *k* (Fig. 2), but some improvements have been included to adapt it to its specific use at the probabilistic estimator.

These improvements are designed to achieve the robustness needed in the segmented data by the probabilistic tracker, and are mainly the following:

1) The cluster updating: Instead of assigning randomly the initial centroides to find the cluster organization, these are obtained from the previous segmentation process through an updating step that uses the model presented in paragraph II (the cluster movement can be estimated calculating its centroide dynamics). With this procedure the algorithm is faster as the clusters are slightly predefined at the beginning of the searching action.

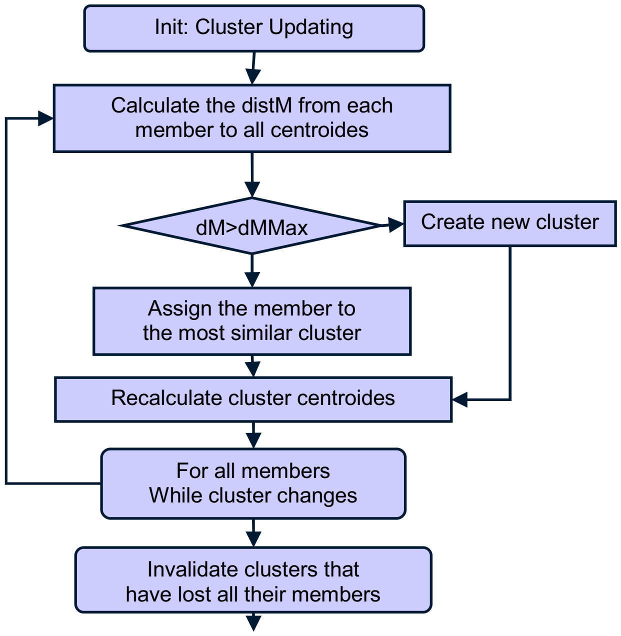

Init: Cluster Updating
Calculate the distM from each member to all centroides
dM>dMMax
Create new cluster
Assign the member to the most similar cluster
Recalculate cluster centroides
For all members While cluster changes
Invalidate clusters that have lost all their members

Fig. 2. Description of the *kmeans* clustering.

2) The cluster validation: When a new cluster is created (does not come from an updated one) it is converted into a candidate that will not be used in the probabilistic algorithm until it is possible to follow its evolution after a programmable number of times. The same process is used to erase a cluster, when it is not validated with new measurements also for a specific moment. This method ensures the robustness of the probabilistic estimator against spurious measurements.

B. The effect of the clustering in the probabilistic tracker

A similar solution to the extended PF for tracking multiple obstacles was proposed in [9]. But that algorithm had a problem of robustness with low accurate measurements, as it has already been mentioned.

The proposed segmentation is inserted to increase this robustness, as it was exposed in the previous paragraph. On the other hand, an environmental adaptive multimodal estimator can be implemented with this information.

1) At the re-initialization stage: With a cluster organization it is possible to select the measurements to be inserted in the prior distribution $p(\vec{a}_{t-1}|\vec{c}_{1:t-1})$ at the re-initialization stage, according to their object assignment (in general *M/k* measurements from each cluster):

$$M = \sum_k m_i \text{, where } m_i = l(k_i) \qquad (6)$$

As newly inserted particles are chosen randomly from groups with high-level concentration of measurements, the likelihood of new objects is high from the beginning. This fact prevents from situations in which particles related to poorly sensed objects are erased from the multimodal distribution at the resampling stage.

The *M/k* particles to be inserted from each cluster are completed with some others randomly selected from its history buffer, which contains measurements assigned to each cluster in previous time steps and that are not very distant from its current centroide. New particles taken from the buffer make the estimation more stable.

2) At the importance sampling stage: The cluster structure is used to obtain a new likelihood function in which each particle is compared to the centroide of the closest cluster:

$$u = \min_c \left\{ \sqrt{\left(\vec{c}_t^{\,i} - \vec{c}_t^{\,c}\right)^2} \right\} \text{ in (5), where } \vec{c}_t^{\,c} = h(\vec{a}_t^{\,c}, \vec{o}_t) \quad (7)$$

With this method the predicted sample set S' is going to be very close to the real state vector, obtaining high values for the likelihood function $p(\vec{c}_t|\vec{a}_{1:t})$ at the importance sampling stage, and thus improving the estimator robustness.

3) At the resampling stage: The cluster information can be used to do a dynamic assignment of the 'M-N' particles to resample among the k different clusters detected, and according to their likelihood too. This fact also prevents from the situations of objects poorly measured whose related particles are erased from the posterior, as mentioned before in the paper.

4) At the output stage: The segmentation is executed again at the end of the extended PF, and this occasion over the S'' sample set, using the centroides of the obtained clusters as the final estimated state vector.

Fig. 3 shows the functionality of the extended PF designed, including the segmentation.

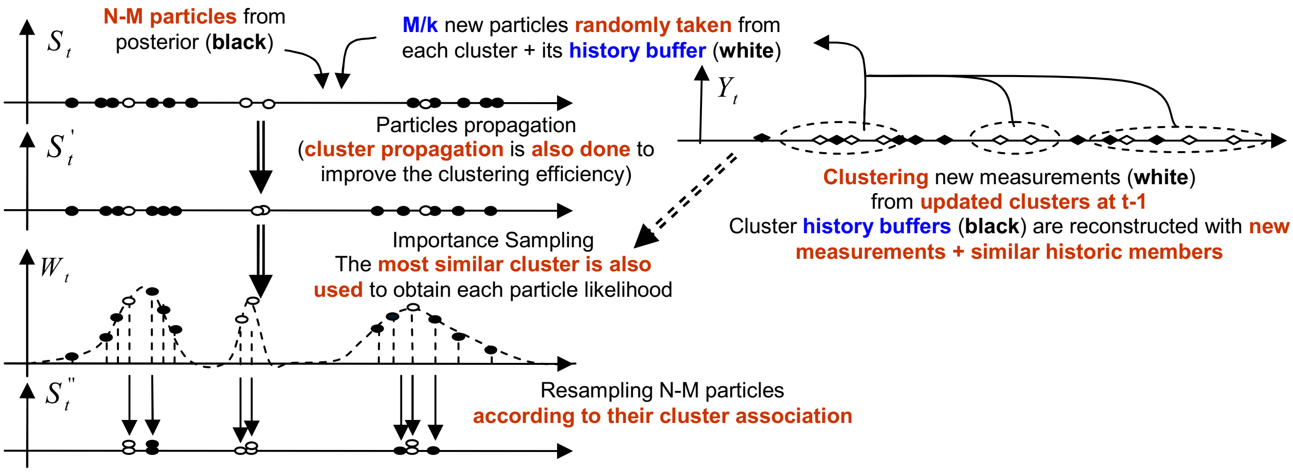

S_t
N-M particles from posterior (black)
M/k new particles randomly taken from each cluster + its history buffer (white)
Y_t
S'_t
Particles propagation (cluster propagation is also done to improve the clustering efficiency)
Clustering new measurements (white) from updated clusters at t-1
Cluster history buffers (black) are reconstructed with new measurements + similar historic members
W_t
Importance Sampling The most similar cluster is also used to obtain each particle likelihood
S''_t
Resampling N-M particles according to their cluster association

Fig. 3. Description of the final proposed estimation algorithm.

V. The Results

To show the performance of the XPFCP, some results are presented in this paragraph, both with sonar data and with stereovision data.

The different parameters used for each experiment will be specified in each case, but the meaning of the plots showing the results is the same in both cases.

A. Results with sonar

Fig. 4 shows the results of one of the tests, in a sequence of images from a) to h). At each of the images there are two plots, a first 3D plot on the top of the image shows the likelihood at each position around the robot (represented by the cylinder in the centre of the plot). The other plot at each image is a 2D view of the particles position colored to identify its cluster association.

The test presented here corresponds to a situation in which there are two static obstacles and a moving one that appears from the right and crosses the robot in the front.

This tests has been done with *N*=200 total number of particles, *M*=30 new inserted particles at the re-initialization phase, a cluster validation parameter of 2 and an estimator sampling rate of 50ms.

The execution time of the XPFCP itself is around 1ms, but as measurements come very sub-sampled (as mentioned in paragraph II in this paper), the sampling rate of the global estimator has been increased, trying not to affect with this modification the maximum detectable speed of the obstacles around the robot.

B. Results with vision

Fig. 5 shows the result of one the tests with vision data. As it can be noticed, frames have been taken from an outdoor environment, but thanks to the filtering process, these measurements are robust enough for the XPF to track the objects in the environment.

The meaning of the two plots in the images from a) to h) is the same of Fig. 4. In this case, there are two obstacles, that cross their path in the experiment, and the tracker follows correctly the position of each one.

The test has been done with *N*=600 total number of particles and *M*=360 new inserted particles al the re-initialization step. This means that both the total number of particles and the relation $\gamma = {}^{M}\!/_{N}$ has been increased. This fact is due to the bigger amount of measurements that are available with this sensory system.

As *N* has increased, the algorithm execution time is in this case around 6ms, which is still a suitable value for a real-time application. Again, the sampling rate of the global process is 50ms, which includes the capture, the filter and the XPFCP execution.

Comparing the results from Fig. 4 and Fig. 5, what it is interesting to notice is that the stereovision follows better the moments of occlusion between the objects in the environment, as the sensor point of view is bigger in this case than if ultrasonic sensors are used.

VI. Conclusions and Future Works

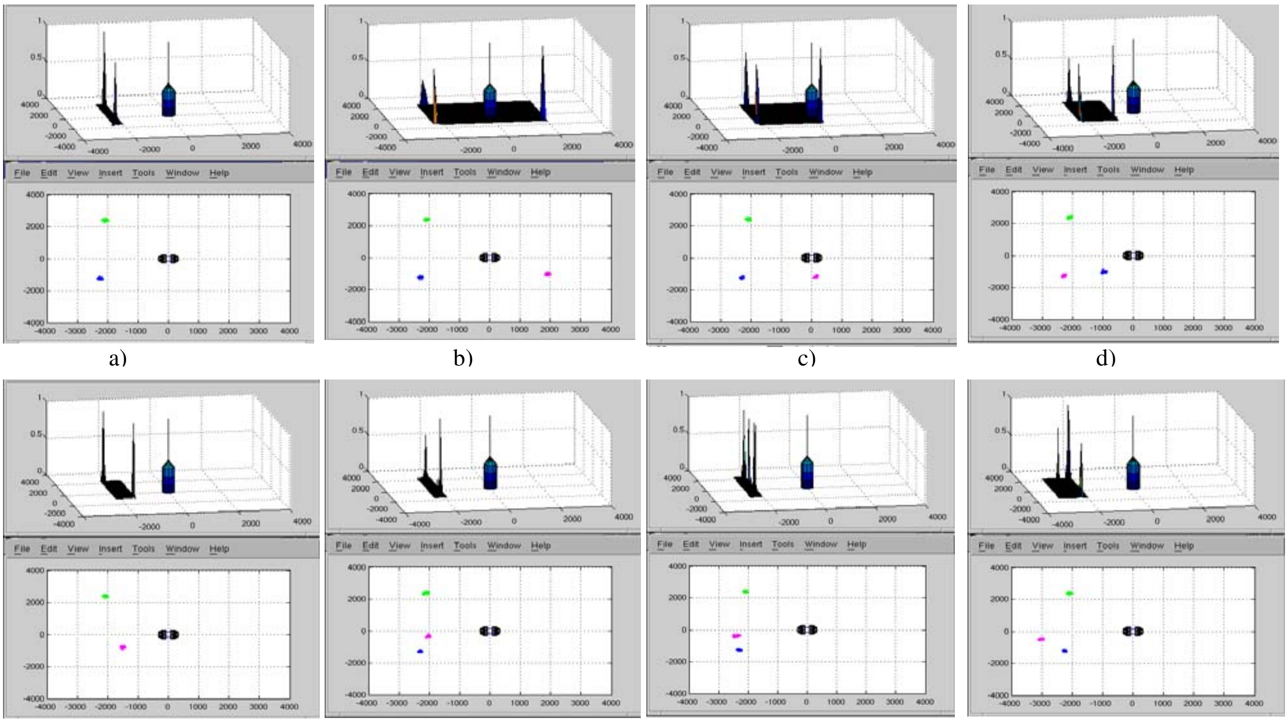
A robust estimator of the movement of the obstacles in the local environment of an autonomous robot has been designed and tested.

The proposed XPFCP is based on a probabilistic multimodal filter, and is completed with a pre-clustering process, obtaining high accuracy and robustness in the tracking task in complex environments.

The segmentation improves the likelihood of the new appearing obstacles, increasing the robustness of a standard multimodal estimator, proposed in other works.

On the other hand, the cluster validation parameter permits an adaptation of the algorithm to the environmental conditions.

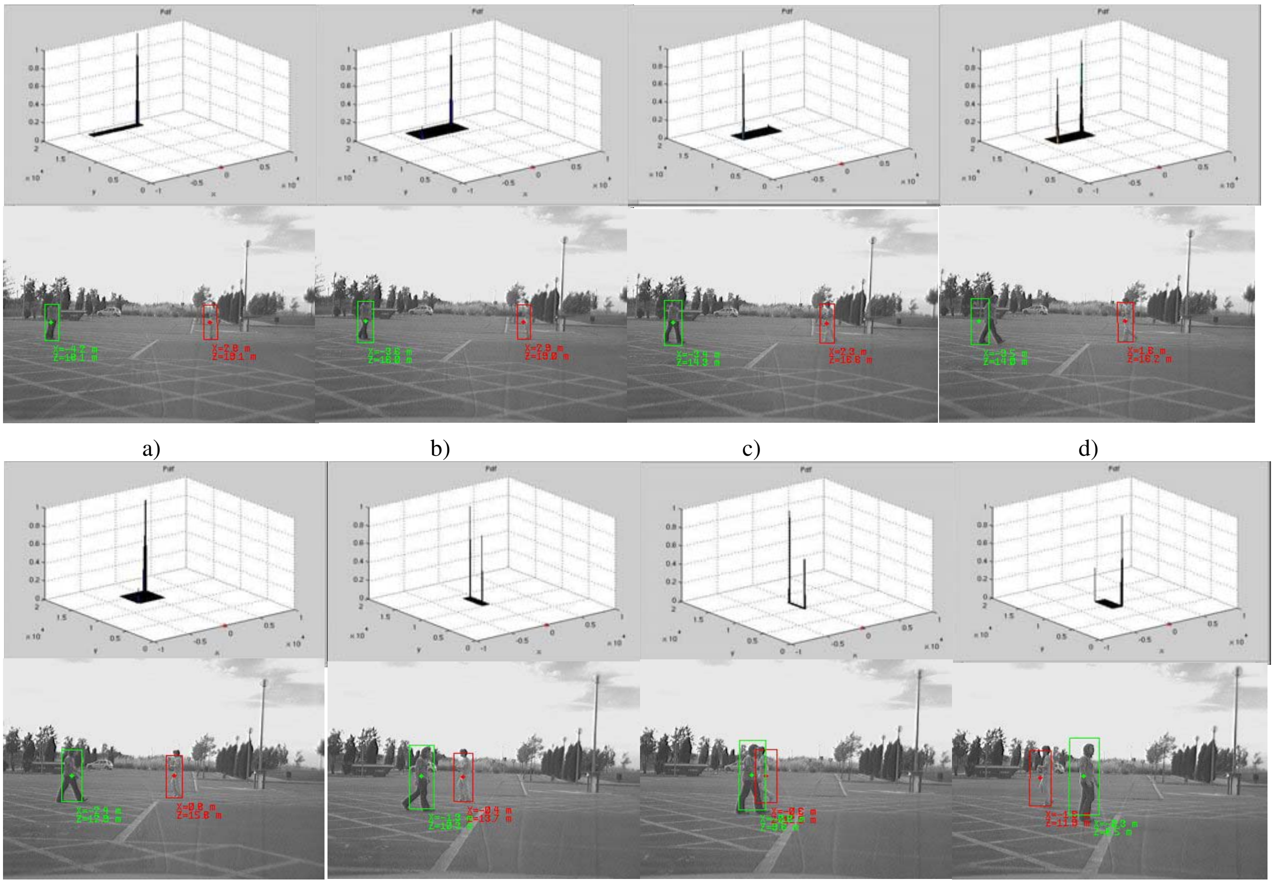
Finally, the execution time of the XPFCP (from 1 to



a) b) c) d)

e) f) g) h)

Fig. 4. Sequential images of a real time experiment with sonar data.



a) b) c) d)

e) f) g) h)

Fig. 5. Sequential images of a real time experiment with stereovision data.

15ms, depending on the type of data that measures the obstacles position) is low enough to allow a real-time tracking in complex environments, and with fast objects. This execution time is not available with some other probabilistic trackers as mentioned in the introduction.

Some future works are still under development. The XPFCP is not totally adapted to the stereovision data, and some improvements have been thought to take advantage of this kind of information.

On the other hand, the XPFCP can easily fuse data up from different kinds of sensors, making the final application more flexible and adaptive to different environmental conditions.

References

[1]. D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "MonteCarlo Localization. Efficient Position Estimation for Mobile Robots" *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, 1999.

[2]. W. Burgard et al., "Experiences with an Interactive Museum Tour Guide", *Journal Artificial Intelligence,* vol. 114(1), pp. 3-55, 1999.

[3]. M. Isard and A. Blake, "Condensation: Conditional Density Propagation for Visual Tracking", *International Journal of Computer Vision,* vol. 29 (1), pp. 5-28, 1998.

[4]. M. Isard and A. Blake, "ICondensation: Unifying Low-Level and High-Level Tracking in a Stochastic Framework", *Proc. 5th European Conference in Computer Vision,* vol. 1, pp. 893-908, 1998.

[5]. M. Orton and W. Fitzgerald, "A Bayesian Approach to Track Multiple Targets Using Sensor Arrays and Particle Filters*", IEEE Transactions on Signal Processing,* vol. 50(2), pp. 216-223, February 2002.

[6]. M. Isard and J. MacCormick, "Bamble. A Bayesian Multiple-Blob Tracker", *Proc. of the International Conference in Computer Vision,* vol. 2, pp. 34-41, 2001.

[7]. D. Schulz, W. Burgard, D. Fox and A.B. Cremers, "Tracking Multiple Moving Objects with a Mobile Robot"*, Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* 2001.

[8]. M. Marron, M.A. Sotelo and J.C. García, "Tracking Multiple and Dynamic Objects with an Extended Particle Filter and an Adapted K-Means Clustering Algorithm", *Preprints of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV),* July 2004.

[9]. E.B. Koller-Meier and F. Ade, "Tracking Multiple Objects Using a Condensation Algorithm", *Journal of Robotics and Autonomous Systems,* pp. 93-105, 2001.

[10]. G. Olague, R. Ramírez, "Síntesis de Imágenes a partir de Fotografías", *DYNA 133*, July, 2001.

[11]. A. Doucet, J.F.G. de Freitas, and N.J. Gordon, *Sequential MonteCarlo Methods in Practice.* Springer-Verlag, 2000.

[12]. R. Van der Merwe, A. Doucet, N. de Freitas and E. Wan, "The Unscented Particle Filter", *Advances in Neural Information Processing Systems 13,* November 2001.

[13]. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp "A Tutorial on Particle Filters for Online Nonlinear Non-Gaussian Bayesian Tracking", *IEEE Trans. Signal Processing,* vol. 50(2), pp. 174-188, February 2002.

[14]. T. Kanungo, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24(7), 2002.