

“XPFCP”: An extended Particle Filter for tracking multiple and dynamic objects in complex environments*

Marta Marrón, Juan C. García, Miguel A. Sotelo, David Fernández and Daniel Pizarro

*Department of Electronics,
University of Alcalá
Edificio Politécnico
Alcalá de Henares, Madrid, SPAIN
marta@depeca.uah.es*

Abstract - The work described in this paper explores a new solution for tracking multiple and dynamic objects in complex environments. An XPF (Extended Particle Filter) is used to implement a multimodal distribution that will represent the most probable estimation for each object position and velocity. A standard PF (Particle Filter) cannot be used with a variable number of obstacles; some other solutions have been tested in different previous works, but most of them require heavy computational resources at least for a high number of obstacles to be tracked. The solution described here includes a clustering procedure that increases the robustness of the probabilistic process in order to provide on-line adaptation to the variable number of clusters. The result is the XPFCP: Extended Particle Filter with Clustering Process. The presented algorithm has been tested using stereovision measurements; the results included in the paper show the efficiency of the proposed system.

Index Terms – Multimodal estimator, tracking, probabilistic algorithms, extended Particle Filter.

I. INTRODUCTION

Probabilistic algorithms have shown their reliability in solving estimation problems since more than fifty years ago. These methods were soon applied to localization problems in autonomous robot navigation ([1] [2]).

PFs (Particle Filters) were developed in the 1990s as the way to implement a discrete distribution and to develop a weighed sampling representation of the Bayesian filter, in a computational cheap fashion, in the framework of the research carried out by Isard and Blake ([3][4]).

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

At this point, JPDAFs (Joint Probabilistic Data Association Filters) have exhibited the highest reliability ([7]). These techniques use a standard PF, or a Kalman Filter, to track each single object. Conversely, the complexity increases when dealing with a dynamic number of objects, requiring a high computational cost that depends on the number of objects under tracking.

The challenge of the XPFCP (Extended Particle Filter with Clustering Process) presented here is to use a single multimodal distribution in order to model the different obstacles in the environment. By doing so, the global

execution time will be made constant and lower than by using the previously mentioned techniques.

The proposed algorithm was firstly described in [8]. It is based on XPFs (Extended Particle Filters) for multiple obstacles [9]. A clustering process has been added to the XPF so as to increase the robustness of the estimator presented in [9].

The algorithm described in [8] is based on an adapted standard PF that dynamically inserts particles from new detected objects and includes a clustering process. The functionality of the XPFCP was presented in [8], where the system was implemented using sonar measurements

In this paper, sound improvements have been made, and the resulting XPFCP performance is shown. Stereovision data are used to track obstacles in dynamic indoor environments. All real results achieved up to date are described.

II. THE MODEL

A. The model equations

The main objective of the XPFCP is to model the movement of the objects that surround an autonomous navigation platform in a complex environment.

In order to achieve this goal, the different detected objects have been characterized by a dynamic model, in which the state vector includes the obstacles relative position and speed, in Cartesian coordinates.

The following equations show the model used to estimate the position and dynamics of the objects:

$$\bar{a}_t = \begin{bmatrix} x_t \\ z_t \\ y_t \\ \dot{x}_t \\ \dot{z}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & f_1() & 0 & 0 \\ 0 & 0 & 0 & f_2() & 0 \\ 0 & 0 & 0 & 0 & f_2() \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ z_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{z}_{t-1} \end{bmatrix} + \bar{r}_t \quad (1)$$

$$\bar{c}_t = \begin{bmatrix} x_t \\ z_t \\ y_t \\ \dot{x}_t \\ \dot{z}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ z_t \\ y_t \\ \dot{x}_t \\ \dot{z}_t \end{bmatrix} + \bar{o}_t \quad (2)$$

where:

* This work has been possible thanks to Spanish Ministry of Science and Technology through PARMEI project (ref. DIP2003-08715-C02-01), and to the University of Alcalá through the ISUAP project (ref. PI2004/033).

- $\begin{bmatrix} x_t & z_t & y_t \end{bmatrix}'$ is the position vector in Cartesian coordinates. The position vector is relative to the centre of the navigation platform. x and z axes conform the moving plane for the robot, and y represents the height. \dot{x} and \dot{z} represent the objects' movement in x and z axes respectively. The speed in y axis is not considered, as the height of any element in the environment is considered to be constant.
- T represents the sampling time of the estimation process.
- $f_1()$ determines the height of the element being tracked by the estimator. The function basically limits the height of the object being tracked in order to increase the robustness and the complexity of the estimator.
- $f_2()$ is used to modify the velocity of the object under tracking. This function depends on the speed of the clusters identified in the clustering process.
- \vec{r}_t and \vec{o}_t are the noise vectors related to the state vector (representing the accuracy of the system model itself) and the output vector (representing the accuracy of the measurements), respectively.

The values of these two vectors are very important for the evolution of the estimator, as \vec{r}_t determines the spread capability of the particles that identify each object, and \vec{o}_t is used in the equation that yields the 'a posteriori' probability of the particle set. Some empirical work was needed to adjust them.

B. Adapting vision measurements to the proposed model

To obtain position measurements of the objects that surround the robot, a stereovision system has been used. A vision process is implemented to extract position measurements (3D coordinates) from 2D images. It is based on the epipolar geometry between the two cameras located in the testing platform.

The vision system is built upon two monochrome digital cameras, synchronized by a Firewire connection, and located in an arrangement onboard a mobile robot with a 30cm baseline.

A classification algorithm has been designed to organize data from the vision procedure into two different groups: data from the obstacles that are the objective of the probabilistic tracker (these elements will be called 'obstacles'), and data from the objects that compose the static environment (that will be called 'objects').

On the one hand, output data from the classifier output belonging to obstacles will be used by the probabilistic estimator for tracking the dynamics of this kind of elements in the environment. On the other hand, data belonging to objects are stored in a file for other possible purposes.

Fig. 1 shows the functional description of the classifier, which includes the procedure implemented to extract 3D points from the 2D left and right images. The third dimension (height = y) is used in this work to filter the possible noise from the 2D images and the epipolar matching and to increase the robustness of the clustering process.

The combination of the camera plus the matching algorithm exhibits an important noise in stereovision measurements. In this application, the noise \vec{o}_t has been characterized as a Gaussian distribution with $\mu = 0$ (white noise) and $\sigma = 50$ mm, after the adequate calibration. The sampling rate of the stereovision sequences has been tested from 66ms (15fps) to 33ms (30fps) with satisfactory results.

The execution time of the classification algorithm is around 60ms, so a real time data acquisition system for stereovision can be carried out using the proposed procedure.

Fig. 2 shows the output of the classifier in one of the experiments along 6 frames of the global sequence. The results for each experiment are displayed in 3 vertically organized images:

- The top image shows the output of the Canny filtering process, where both obstacles and objects edges are mixed in the image.
- The picture in the middle one shows the result of the classification process in the XY projection (x range is from -8m to 8m, and z range is from 0.5m to 16.5m). White dots represent the measurements that, according to the classifier results, belong to obstacles.
- The bottom one shows the left image of the associated frame, where the points detected as obstacles have been plotted in their relative XYZ position.

Once the classification process has been performed the resulting obstacle data set can be inserted in the probabilistic estimator.

III. THE EXTENDED PARTICLE FILTER

In these paragraphs, the functionality of the proposed XPF is explained. A more complete description of the algorithm can be found in [8].

A. The XFP

The main loop of a standard PF ([10] [11] [12]) starts at time t with a set $S_t = \{s_t^i / i = 1..N\}$ of random particles representing the posterior distribution of the state vector $p(\vec{a}_{t-1} | \vec{c}_{1:t-1})$ estimated at the previous time step.

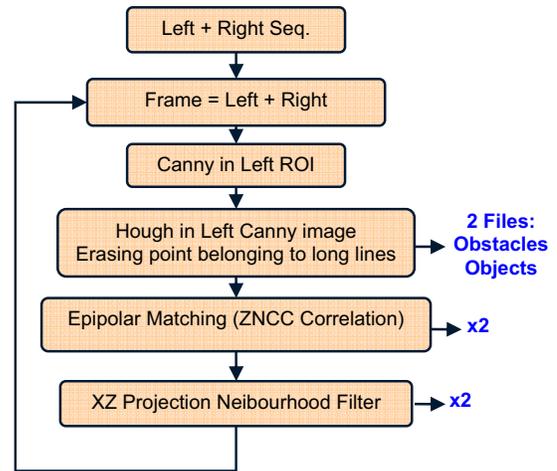


Fig. 1. Functional description of the vision-based classification algorithm.



Fig. 2. Results of the classification algorithm.

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

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

Using the weighing vector, and applying the selected resampling scheme, a new set S''_t is obtained with the most probable particles, which will represent the new $p(\bar{a}_t | \bar{c}_{1:t})$.

The standard PF estimates quite well the evolution of any kind of single object defined by its model, but as mentioned in [9], two constraints make it useless in order to track multiple and dynamic objects:

- The standard PF is not able to estimate new appearing objects in the environment because the measurements related to new objects would be rejected at the resampling stage. To solve this problem, a re-initialization phase is included in the PF standard loop to insert new particles directly from the measurements in the sample set. The resampling step also needs to be modified so as to allow the incorporation of the new particles.
- In order to track multiple objects, the importance sampling step should 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 new object can also be rejected as mentioned in the previous paragraph.

Fig. 3 presents a graphical description of the XPF. The functionality of the modified re-initialization, importance sampling, and resampling methods is detailed in the following paragraphs.

B. The effect of the XPF tracking multiple objects

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

With this modification, the distribution $p_m(\bar{a}_t | \bar{c}_{1:t})$ from the new environmental configuration is combined with the posterior distribution $p_p(\bar{a}_t | \bar{c}_{1:t})$ to obtain a new expression according to (3):

$$p(\bar{a}_{t-1} | \bar{c}_{1:t-1}) = \gamma_t \cdot p_m(\bar{a}_{t-1} | \bar{c}_{1:t-1}) + (1 - \gamma_t) \cdot p_p(\bar{a}_{t-1} | \bar{c}_{1:t-1}) \quad (3)$$

where γ_t is the factor that weighs up the distribution association. It is fixed by the relation between the M_t samples directly inserted from the output vector measured at $t - 1$ and the total number of samples (N) in the particle set ($\gamma_t = M_t / N$).

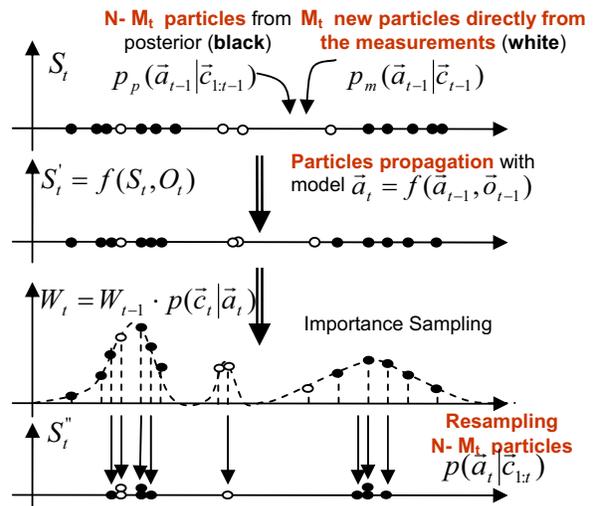


Fig. 3. Description of the XPF functionality.

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

It must be remarked that the reinforcement of the new obstacles being tracked is not ensured if measurements are randomly selected from all the input data. This problem can be solved by doing an oriented selection of the particles to be inserted at the re-initialization phase. The organization of the necessary measurements is achieved thanks to the proposed clustering process.

2) *At the importance sampling stage:* To enable the possibility of representing multiple objects in a single distribution, the particle weights at the importance sampling step have to be obtained according to the specific object that is being tracked by each particle:

To achieve this goal, the base function used to obtain the likelihood is modified as follows:

$$p(\bar{c}_t | \bar{a}_t) = e^{-d_t^2 / 2\sigma^2} \quad (4)$$

where:

$$d_t^i = \min_{m_i} \left\{ \sqrt{(\bar{c}_t^i - \bar{c}_t^j)^2} \right\} \quad (5)$$

As shown in (5), the new distribution is obtained from the minimum distance from each particle \bar{c}_t^i to all measurements \bar{c}_t^m at each time step.

Function d_t has the problem (as commented in [9]) of giving more weight to the objects that are measured with higher accuracy, probably rejecting particles related to the most poorly sensed objects. The clustering process included in the final XPFCP will also solve this problem.

3) *At the resampling stage:* To insert the new M_t particles, the resampling step is also modified. In this case, only $N - M_t$ samples have to be selected from the N existing particles in the S_t^i sample set to generate S_t^r .

Again, the proposed clustering process can be used to selectively choose the particles to be resampled according to their cluster association.

IV. THE CLUSTERING PROCESS

A. The modified *k-means* algorithm

A segmentation procedure has been designed to organize the measurements that come from the stereovision data (previously classified as obstacles' data) in a variable number of clusters.

This method is based on a standard '*k-means*' algorithm ([13]) with unknown initial number of clusters k_t (Fig. 4). Some improvements have been included to adapt it to its specific use in the probabilistic estimator. The improvements are designed to enhance the probabilistic tracker robustness, and are mainly the following:

1) *Cluster updating:* Instead of randomly selecting the elements that conform the initial centroids to find the cluster organization, these are obtained from the previous segmentation phase by means of an updating step that relies on the model (1) (cluster movements can be estimated by calculating its centroid dynamics). Using this procedure the

algorithm is faster as long as the clusters are slightly predefined at the beginning of the searching action.

2) *The cluster validation:* Once a new cluster is created, it is converted into a candidate that will not be used in the probabilistic algorithm until it will be possible to follow its dynamic evolution. This is achieved after a certain number of iterations. The same procedure is used to erase a cluster when it is not validated by new measurements after a specific number of iterations. This method ensures the robustness of the probabilistic estimator against spurious measurements.

B. The effect of the clustering process in the probabilistic tracker

A similar solution to the XPF for tracking multiple obstacles was described in [9]. That algorithm has a problem of robustness with low accurate measurements.

The proposed clustering process is inserted aiming at increasing robustness. On the other hand, an environmental adaptive multimodal estimator can be implemented relying on this information.

The effect of the segmentation in the XPF is described in the following paragraphs:

1) *At the re-initialization stage:* Based on the cluster organization, it is possible to select the measurements to be included in the prior distribution $p(\bar{x}_{t-1} | \bar{y}_{1:t-1})$ at the re-initialization step according to their object assignment:

$$M_t = \sum_{k_t} m_t^i \quad (6)$$

where:

$$m_t^i = f(k_t, cluster_t^i) \quad (7)$$

The tracking probability of new objects is high from the beginning, since inserted particles are randomly chosen from groups that exhibit high-level concentration of measurements.

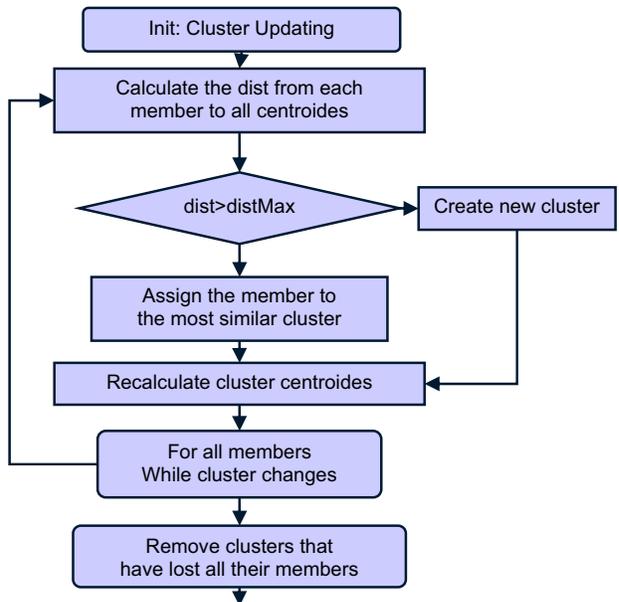


Fig. 4. Description of the '*k-means*' clustering algorithm.

This fact prevents situations in which particles related to poorly sensed objects are removed from the multimodal distribution at the resampling phase.

The m_t^i particles to be inserted from each cluster are completed with some others, randomly selected from its history buffer. History buffers contain measurements assigned to each cluster in previous time steps as long as they are not very distant from the current cluster centroid.

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 closest cluster centroid:

$$d_t^i = \min_{k_i} \left\{ \sqrt{(\bar{c}_t^i - \bar{c}_t^j)^2} \right\} \quad (8)$$

in (4), where \bar{c}_t^k represents the centroid location for each cluster.

By using this method the predicted sample set S_t'' will be very close to the real state vector, yielding high values for the likelihood function $p(\bar{c}_t | \bar{a}_{1:t})$ at the importance sampling step, and thus improving the estimator robustness.

3) *At the resampling stage:* The cluster information can be used to do a dynamic assignment of the $N - M_t$ particles to be resampled among the k_t detected clusters, according to their likelihood too.

Fig. 5 shows the performance of the final XPCP designed, including the clustering process (XPFCP).

V. THE RESULTS

Fig. 6 displays the result of a stereovision-based test using the XPFCP. As can be observed, the frames have been taken from a complex and unstructured indoor environment.

Thanks to the classification algorithm presented in paragraph II (Fig. 2), only those measurements related to obstacles in the environment are tracked by the probabilistic estimator.

The figure shows the XPFCP output in 10 frames in a global sequence going from a) to j). Each frame has 2 vertically organized images:

- The upper one displays the left image. Green dots represent the measurements (\bar{c}_t in the proposed model

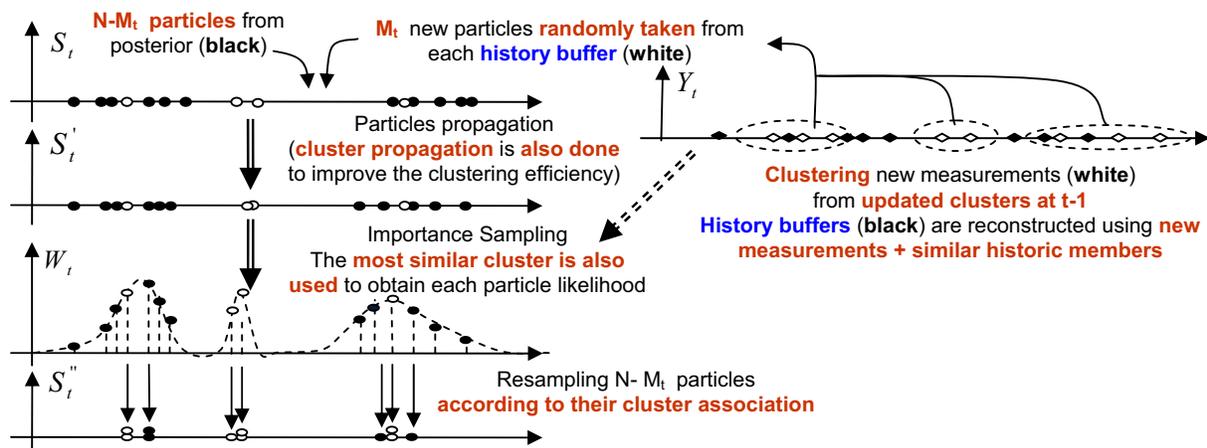


Fig. 5. Description of the final proposed estimation algorithm (XPFCP).

(2)), red dots represent the particles' positions (part of \bar{a}_t (1)), and the square represents the output. Cluster shapes (modelled as vertical cylinders in the 3D world) show the final and deterministic number and position of obstacles in the environment.

- The lower image presents the XY projection of the 3D results shown in the upper figure. The meaning of the different elements that appear in this plot is the same as the one explained in the previous paragraph. In this case, a circle is the corresponding shape of the cylindrical model at the end of each XPFCP execution loop.

The test has been done with a total number of $N = 600$ particles and $\gamma_t \approx 0.6$ new inserted particles at the re-initialization step. As the total number of particles is kept constant, the XPFCP execution time is also constant, with a value of around 18ms. Therefore, the sampling time of the global application (classification process and XPFCP algorithm) will be limited by the first one, to 60ms. It implies a performance of 15fps in the final tracking system. This value is more than enough for a real time tracking system in indoor environments.

The results displayed in Fig. 6 show that the tracker follows correctly the position of each obstacle.

Some problems appear with occlusions. In the experiment shown in Fig. 6, only h) and i) situations depict this problem. Some proposals are already being tested in order to solve it, as mentioned in the next paragraph.

VI. CONCLUSIONS AND FUTURE WORK

A robust estimator of obstacles movement in unstructured, indoor environments has been designed and tested.

The proposed XPFCP is based on a probabilistic multimodal filter that is completed with a clustering process, obtaining highly accurate and robust multitasking in complex environments.

A specific classification algorithm for stereovision data has also been developed. This process is able to separate vision measurements belonging to obstacles from those belonging to objects that are part of the environment, simplifying the XPFCP dynamic estimation task.

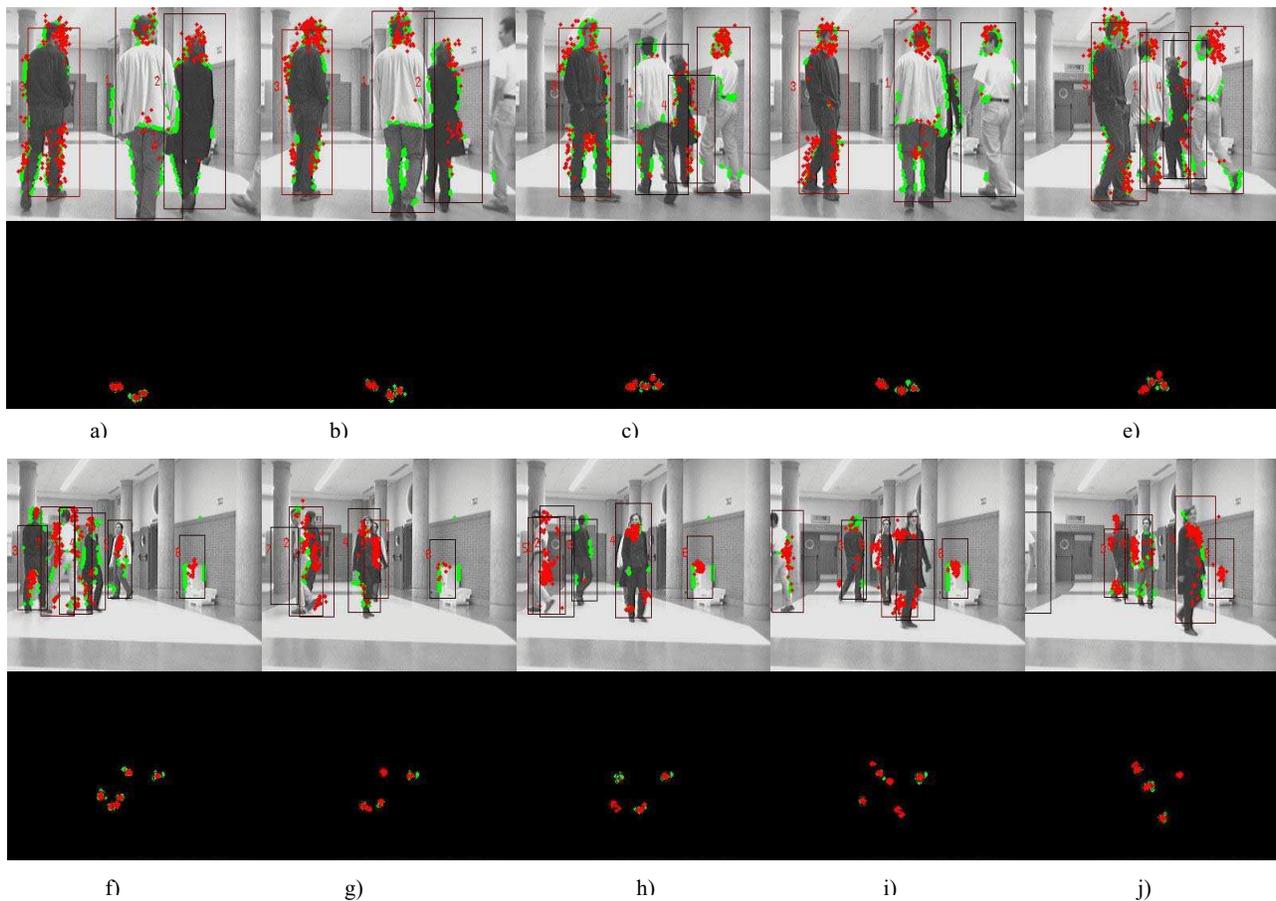


Fig. 6. Sequential images of a real time experiment using stereovision data.

The clustering process improves the probability of new appearing obstacles, increasing the robustness of a standard multimodal estimator, proposed in other solutions. A subtractive clustering algorithm is being tested in order to improve the behaviour of the estimator in occluded situations.

In addition, the execution time of the global application is low enough to allow for real-time tracking of fast objects in complex environments.

Finally, the XPFCP designed can easily fuse data coming 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 on 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, "Bamle. 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] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, *Sequential MonteCarlo Methods in Practice*. Springer-Verlag, 2000.
- [11] 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.
- [12] 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.
- [13] T. Kanungo, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24(7), 2002.