IECON 2005

The Thirty-First Annual Conference of the IEEE Industrial Electronics Society

Sheraton Capital Center
Raleigh, North Carolina, USA

6-10 November 2005

Sponsored by
IEEE Industrial Electronics Society

Hosted by
North Carolina State University

Edited by
Leopoldo G. Franquelo
Alexander Malinowski
Mo-Yuen Chow
Herbert L. Hess

IEEE Catalog Number: 05CH37699

ISBN: 0-7803-9252-3

ISSN: 1553-572X

Additional Copies of this Publication are Available from:

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08854-1331
Phone: (800) 678-IEEE
(732) 981-1393
Fax: (732) 981-9667
E-mail: customer.services@ieee.org

MR-P1: Mechatronic Systems

Room: Mezzanine, Wednesday, Nov 9, 10:15 to 11:30 AM
Session Chairs: Karel Jezernik

Clustering methods for 3D vision data and its application in a probabilistic estimator for tracking multiple objects

Marta Marrón, Juan C García, Miguel A Sotelo, Emilio J Bueno
Electronics Department. University of Alcala
Edificio Politécnico. Campus Universitario. Alcalá de Henares
Madrid. 28871
SPAIN
{marta,jcarlos,sotelo,emilio}@depeca.uah.es

Abstract **–Probabilistic algorithms have been fully tested as the best solution in multiples areas, and thus in tracking tasks. Different solutions with them have been proposed for multiple objects tracking. The proposal of the authors is based on a particle filter whose robustness and adaptability is increased by the use of a clustering algorithm. Two different proposals for the segmentation process are presented in this paper, and interesting conclusions are extracted from their functional comparison. Tracking results are also presented in the paper, showing the reliability of the proposals.**

I. INTRODUCTION

Probabilistic algorithms in their different forms (Multi-Hypothesis Techniques -MHT- [1], Particle Filters -PF- [2] and their diversifications [3][4]) have fully shown their reliability in estimation objectives. Nowadays these methods are widely applied solving positioning problems in robot autonomous navigation ([5] [6]).

The idea of tracking multiple objects appeared with the first autonomous robot, and soon probabilistic algorithms were applied to achieve this aim ([7] [8] [9]). The multiplicity problem is not easily solved with an optimal estimator such as a Kalman Filter, although the model in use accomplishes the linearity constraints.

An initial solution for the multiple objects tracker is to use a standard PF to track each object but this is not efficient as it does not work with a dynamic number of objects. Some other solutions include another estimator (a stochastic one in most of cases) to obtain the association among the measurements and the particles of the filter (Joint Probabilistic Data Association Filter -JPDAF-) [10], but the computational cost of these techniques is high.

A different solution to achieve a multiple objects tracker has already been developed by the authors in [11]. The algorithm has been tested in complex indoor environments with sonar and stereovision data with good results. The method is based on a standard PF slightly modified to dynamically include particles from the new objects. The result is the Extended Particle Filter (XPF). A clustering application is added in this point to organize the deterministic and stochastic information of the estimation process, and to increase with the segmented data its robustness and reliability.

In this paper, a slight description of the XPF developed in [11] is firstly included; the description of the two proposed classifiers is presented secondly; some comparisons between the functionality of these two processes are presented after; and finally some results and conclusions obtained from the segmentation algorithm applied to the estimator are exposed at the end.

II. THE PROBABILISTIC ESTIMATOR: XPF

The main objective of the XPF is to model the movement of the objects that surround an autonomous navigation platform in a complex environment. To achieve this aim the different detected objects have been characterized by a dynamics model, whose evolution will be estimated by the modified PF.

In the following paragraphs both the model and the XPF basics will be exposed, though a more detailed information about these two points can be found in [11].

A. The dynamic model

The main specifications of the model used in the XPF for tracking multiple objects are the following:

- *The state vector* $\vec{a}_t$ includes the position and speed in Cartesian coordinates, relative to the robot ones (XZ is the horizontal moving plane and Y is the height dimension in the XZY configuration space used):
$$\vec{a}_t = \begin{bmatrix} x_t & z_t & y_t & \dot{x}_t & \dot{z}_t \end{bmatrix} \quad (1)$$
- *The measurements vector* $\vec{c}_t$ includes only the position information that has been extracted with an stereovision process:
$$\vec{c}_t = \begin{bmatrix} x_t & z_t & y_t \end{bmatrix} \quad (2)$$
- *The model* is linear and the measurements are affected by a Gaussian noise due to the camera itself plus the stereovision algorithm, that after calibration has been identified as a distribution $\sigma = 50$ mm.

B. The XPF functionality

The main loop of a standard PF ([12]) based on the SIR algorithm ([13]) starts at time t with a set $S_t = \{s_t^i / i = 1..N\}$ of N random particles representing the posterior distribution of the state vector estimated $p(\vec{a}_{t-1} | \vec{c}_{1:t-1})$ at the previous step. The rest of the process is developed in three steps, as follows:

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

b) The weight of each particle $W_t = \{w_t^i / i = 1 \ldots N\}$ is then obtained comparing the measured output vector and the predicted one based on the prior estimations. Using the SIR version of the filter, these weights are obtained directly from the likelihood function $p(\vec{c}_t|\vec{a}_t)$, and then normalized.

c) Using the weights vector W, and applying a resampling scheme, a new set S_t'' is obtained with the most probable particles, which will represent 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 it is not able to estimate new appearing objects because the measurements related to them would be rejected at the resampling step and no particles would have enough weigh to represent over time the evolution of the new object.

To solve this problem, a re-initialization step is inserted in the PF standard loop to add M new particles directly from the measurements in the sample set. The resampling step also needs to be modified as only N-M samples have to be selected from the N existing at the S_t' sample set.

On the other hand, 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. 1 presents a graphical description of the XPF explained.

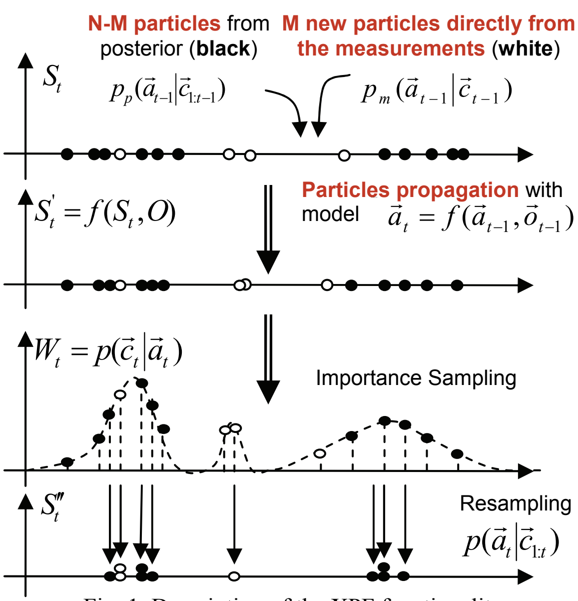

Fig. 1. Description of the XPF functionality.

These considerations, extracted from [14], have shown some problems of robustness, that the clustering process presented by the authors solve, as it was demonstrated in previous communications [11]. The main problems are the following:

- *At the re-initialization step,* if measurements are randomly selected from all the input data, the reinforcement of the new obstacles tracking is not ensured. This problem will be solved with an oriented selection of the M particles to be inserted at the re-initialization phase. The organization of the needed measurements is obtained thanks to the proposed clustering process, as it will be detailed later.
- *At the importance sampling step,* to enable the multi-tracking procedure with a single distribution, the function used to obtain the weights is modified as follows:

$$w_t^i = e^{-\left(d_t^i\right)^2 / 2\sigma^2} \quad \text{where} \quad d_t^i = \min_m \left\{ \sqrt{\left(\vec{c}_t^{\,i} - \vec{c}_t^{\,m}\right)^2} \right\} \quad (3)$$

The function d has the drawback of giving more weight to the particles related to objects measured with higher accuracy, and probably rejecting those related to the most poorly sensed objects. The clustering process included in the final XPF will also solve this problem, as it partially filters measurements inaccuracy.

All these considerations will be analyzed in the following paragraphs, from the point of view of each of the segmentation methods proposed.

III. THE CLUSTERING ALGORITHMS

As it has been remarked in the previous sections, the standard probabilistic estimator is not reliable enough to track robustly multiple targets with an only multimodal density. Some kind of association between the different objects to track and the measurements that come from them is needed. A very spread implemented example of this idea is the JPDAFs that is based on the development of another Bayesian filter to estimate the relation between new measurements and particles.

The proposal presented by the authors is the use of a segmentation algorithm that increases the robustness of the estimation procedure in different points.

In this section, the effects of the clustering algorithm applied to the XPF presented previously are exposed, and on the other hand, the functionality of two segmentation methods tested with the multi-tracker are explained:

- A modified '*k-means*'.
- A modified subtractive.

A. Including a clustering process in the XPF

The main idea in the use of a classifier is to group the measurements that come from the different objects to track, in order to increase the likelihood values that are used afterwards in the estimator.

The use of segmented measurements in the XPF can be applied in the different steps of the probabilistic estimator with different effects in its final robustness:

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

$$M = \sum_k m_j \text{, where } m_j = g(k_j) \quad (4)$$

With this method, the weigh of particles related to new objects is high from the beginning, because they are chosen from groups with high-level concentration of measurements. The *M/k* particles to be inserted from each cluster are randomly selected from a history buffer that contains measurements assigned to each cluster in this and previous time steps and that are not very distant from its current centroide. Taking new particles from the buffer make the estimation more stable.

b) *At the importance sampling step*: The cluster structure is used to obtain a new function to calculate the weights, in which each particle is compared to the closest centroide ($\vec{c}_t^{\,b}$):

$$d_t^i = \min_k \left\{ \sqrt{\left(\vec{c}_t^{\,i} - \vec{c}_t^{\,b}\right)^2} \right\} \quad (5)$$

c) *At the resampling step:* The cluster information can be used to do a dynamic assignment of the *M-N* particles to be resampled among the *k* clusters detected. This fact also prevents from situations of objects poorly measured whose related particles would be in any other case, erased from the posterior distribution.

d) *At the output stage:* The clustering process is executed again at the end of the XPF, but in this occasion over the S_t'' sample set, using the centroide of the obtained clusters as the output deterministic value for the estimated state vector at each time step.

Fig. 2 shows the performance of the final XPF designed, including the classifier. In the following paragraphs the proposed segmentation algorithms are presented.

B. The modified 'k-means' clustering algorithm

The first segmentation method that has been designed to organize the measurements in a variable number of clusters is based on a standard *'k-means'* [14] with unknown initial number of clusters *k* (Fig. 3).

Some improvements have been included in the standard algorithm in order to adapt it to its specific use in the estimator. These are mainly the following:

- *The cluster update:* Instead of selecting randomly the elements that will be the initial centroides to find the cluster organization, these are obtained from the previous segmentation phase through an updating step that uses the model (cluster movements can be estimated calculating its centroide dynamics). With this procedure, the application is faster as the clusters are slightly predefined at the beginning of the searching action.
- *The cluster validation:* When a new cluster is just created, it is converted into a candidate that will not be used in the probabilistic algorithm until it is possible to follow its dynamic evolution after a certain number of times. The same procedure is used to erase a cluster when it is not validated with new measurements, for a specific number of times. This method ensures the robustness of the estimator against spurious measurements. Fig. 4 shows a functional diagram of the validation process.

In the multi-tracking tests done with the XPF and this classifier, the differential characteristic in the cluster's organization has been the Euclidean distance in the XZ plane from the data to the cluster centroides. The limit-distance ('*distMax*') parameter can be defined adaptive or fixed for any type of application.

The same procedure is applied to the cluster validation, where the validation parameters can be defined in two characteristics, and can also be adaptive of fixed:

- *Euclidean distance* between the previous and actual centroide location for each cluster.
- *Number of members and its evolution.* This parameter is considered as an idea of the cluster reliability, and thus its use to validate it or not is very important.

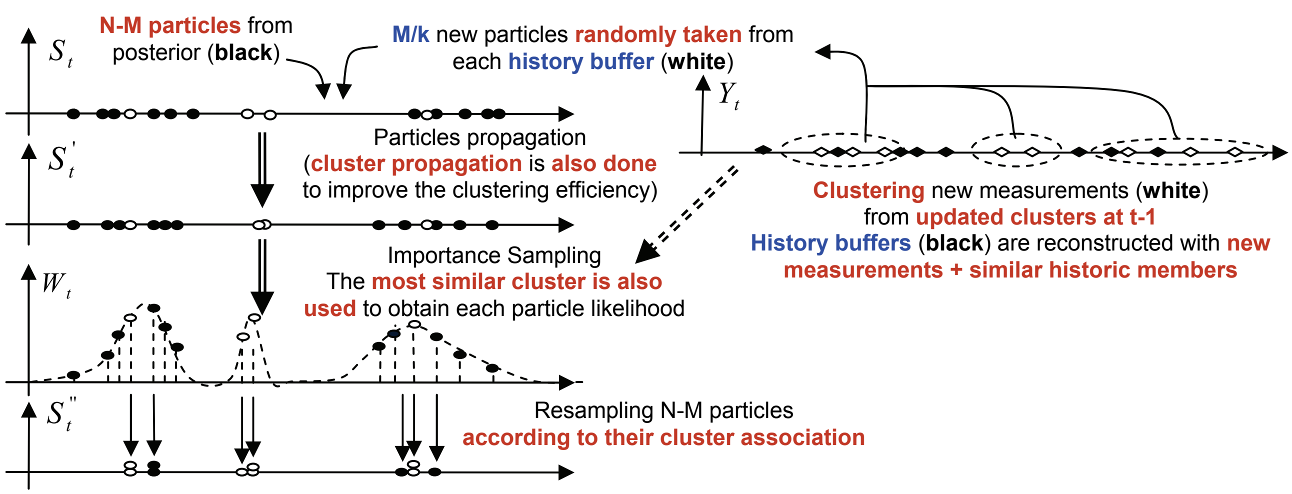

Fig. 2. Description of the final proposed estimation algorithm.

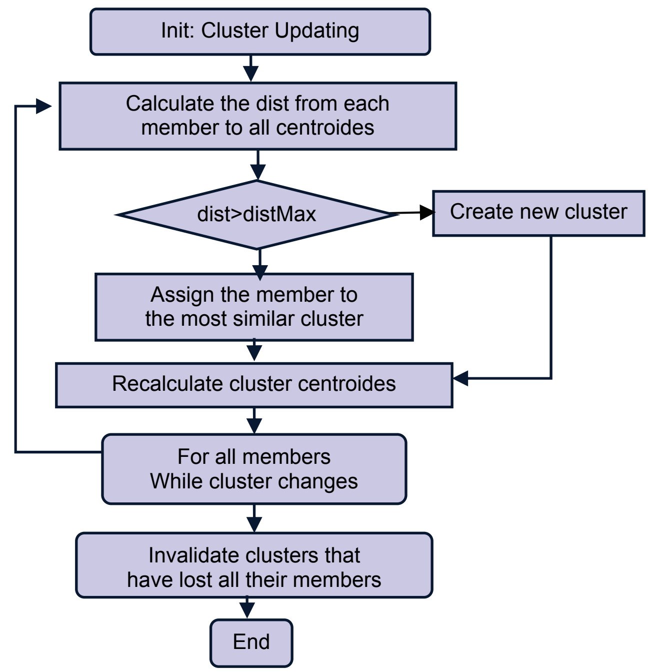

Fig. 3. Description of the *'k-means'* clustering algorithm.

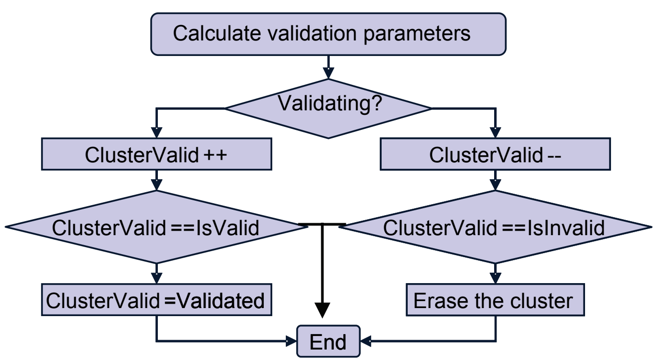

Fig. 4. Description of the validation algorithm.

The validation algorithm increase the robustness of the estimator, as only the validated clusters are used in the re-initialization step of the XPF and in the importance sampling step. This process has a second order effect that reflects on a smoother answer in the estimation task, but its effect is compensated by the robustness of the multimodal density.

Fig 5 show the result in different sequential frames of this classifier applied to the XPF in a test made from stereovision data. Green squares and dots show the measurements segmented at each time step. These clusters will be used in the XPF steps a) to c) as explained in section III.A. Red squares and dots represent the particles segmented at the end of the XPF, that is the deterministic output of the multi-tracker at each time step, as explained in step d) section III.A.

The colour intensity, in any case, show the probability corresponding to the segmented cluster, the blacker it is the less sure the classification is. In all tests shown in the paper the Euclidean distance limit is set to 0.5m, and the validation rate to 3 times. Each frame includes a label in the bottom the number of clusters (labelled as *'k'*) and validated ones (labelled as *'kValid'*), detected at each time step by the segmentation process in the measurements and in the particles as the last step in the XPF (labelled with *'-Out'* suffix).

As it can be appreciated in the figure, the cluster works perfectly in all situations, as so does the final probabilistic estimator proposed (XPF plus classifier). It tracks a new person when it appears, and follows it almost when it totally disappears.

More conclusions will be analyzed in the last section of the paper, comparing these results with the ones obtained with the subtractive method.

C. The modified subtractive clustering algorithm

The second segmentation process, called subtractive [16], designed to incorporate to XPF is a fuzzy modification of the standard *'k-means'*.

This method is based on the probability that each measurement has of being the centroide of a cluster. Its probabilistic soundness makes it especially interesting in the application of this paper. Fig. 6 shows a functional diagram of the standard subtractive method.

This algorithm has some advantages if compared with the previously presented:

- As a fuzzy classifier, the measurements belong to one cluster in a more relaxed way than the *'k-means'*, with a continuous value. This information will be useful in the validation process presented in previous section, and also in the XPF resampling step, as it is explained later in the paper.
- At the end of the segmentation process, not all measurements have to belong to one of the clusters. This fact behaves as a noise filter against spurious measurements, which with the previous method would generate a non-validated new cluster.

The continuous likelihood value that is obtained with the algorithm for each cluster is also used in the XPF resampling step. With this information, the resampling can be applied to reinforce the poorest measured objects, making the estimation more robust.

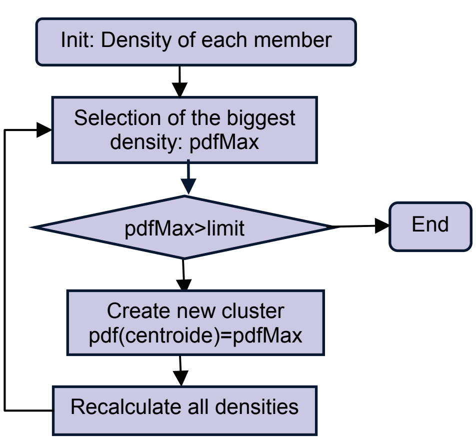

Fig. 6. Description of the subtractive clustering algorithm.

Fig. 5. Results obtained with the XPF and the *'k-means'* clustering algorithm.

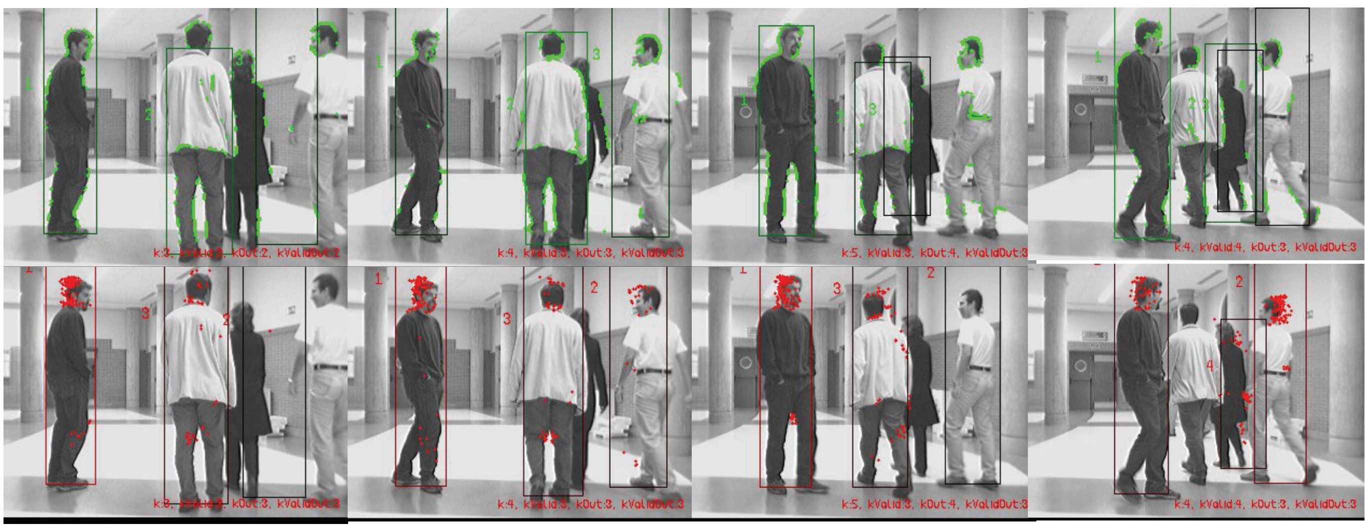

Fig. 7. Results obtained with the XPF and the subtractive clustering algorithm.

On the other hand, the subtractive method has also some drawbacks:

- The global application does not include a step to assign members to clusters. This part of the segmentation will need to be done afterwards with a one pass loop process, and using the Euclidean distance in a similar way as it was used in the *'k-means'* algorithm to do the classification.
- The cluster update step is useless with this classifier, as a global density search has to be done. This fact will make the application more time consuming than the '*k-means'* one, as the only consequence.

The extraction of the density values in the standard subtractive algorithm has been speed up using a discrete grid in its final implementation. The grid size is adaptive to the data available at each time step. Nevertheless, the centroide obtained from the classifier is not directly obtained from the grid, so the continuous value of this variable is maintained.

Finally it is necessary to point out that the validation process is implemented in the same way in the subtractive clustering as it was in the *'k-means'* one.

Fig. 7 shows the results of the XPF in which the subtractive method has been applied. The test bench has been the same to the one used in the results shown in Fig. 5.

The results shown in this case also confirm the functionality of the subtractive clustering, and its application to the XPF. More conclusions related to the comparison between the results shown in Figs. 5 and 7 are discussed in the following section.

IV. COMPARISONS AND CONCLUSIONS

In the experiment shown in Figs. 5 and 7 the following situation is presented in 4 sequential frames:

a) Three persons area being correctly tracked in their way, when a fourth person appears in the scene from the left.

b) The classifier segments in one or two steps the new object (green square) and the probabilistic tracker generates the corresponding cluster in the output (red dots and square).

c) Meanwhile, one of the tracked persons at the beginning of the scene is almost totally occluded by another one, leaving very few measurements from this object to the classifier (green dots).

d) With this situation, the occluded person is very difficult to track, but thanks to the classifier its location and way is recovered with time.

Comparing the results obtained from the two experiments it can be noticed that the subtractive proposal do not give enough likelihood to the cluster related to the occluded person. Thus, the XPF output cluster associated is not finally validated.

As a conclusion it must be pointed out that a more refined configuration of the segmentation process parameters would help in the detection and tracking of such a situation.

V. ACKNOWLEDGMENT

This work has been financed by the Spanish administration (CICYT: DPI2002-04555-C04-04).

VI. REFERENCES

[1] D. Maksarov H. Durrant-Whyte “Mobile vehicle navigation in unknown environments: a multiple hypothesis approach”, *IEE Proceedings in Control Theory & Applications.* vol. 142, 1995.

[2] 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.

[3] D. Fox et al, “Bayesian Filtering for Location Estimation” *Pervasive Computing,* July 2003.

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

[5] 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.

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

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

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

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

[10] 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.

[11] M. Marron, M.A. Sotelo and J.C. García, “Design and applications of an extended particle filter with a pre-clustering process, XPFCP” *Proccedings of the IEEE conference on Mechatronics & Robotics 2004*, pp: 187-191, September 2004.

[12] S. Thrun, “Probabilistic Algorithms in Robotics”, *AI Magazine,* Winter 2000.

[13] M.S. Arulampalam, et al.“A Tutorial on Particle Filters for Online Nonlinear Non-Gaussian Bayesian Tracking”, *IEEE Transactions on Signal Processing,* vol. 50(2), pp. 174-188, February 2002.

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

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

[16] Wen-Yuan Liu, et al. “Study on Subtractive Clustering with fuzzy c-Means Clustering”, *Proc. of the 2nd International Conference on Machine Learning and Cybernetics,* November 2003.