A Particle Filter on Interval Data for Mobile Localization

F. Abdallah, Ph. Bonnifait, A. Gning
Heudiasyc UMR 6599
Université de Technologie de Compiègne
{fahed.abdallah, philippe.bonnifait,gningelh}@hds.utc.fr
Tel.: +33 3 44234473 Fax: +33 3 44234477

1. Short Abstract

Particle filters, or Sequential Monte Carlo Methods are among the most promising candidates to provide solution to the problem of mobile robot localization. This is due to the fact that this kind of problem needs to include elements of non-linearity and non-Gaussianity in order to model accurately the underlying dynamics of the system. Particle filter methods solve the localization problem as a Bayesian filtering problem in order to estimate the posterior density of the state using weighted particles.

In many real applications, information is often imprecise, i.e. biased and noisy. Such an imprecision can be easily represented by interval data if the maximum error is known. Handling interval data is a new approach successfully applied to different real applications.

In this paper, we propose an extension of the particle filter algorithm (called box particle filter) to deal with interval data by using interval analysis and Constraints Satisfaction techniques. In usual particle filtering, particles are punctual states associated with weights whose likelihood is defined by a statistical model of the observation error. In the box particle filter, particles are boxes associated with weights whose likelihood is defined by a bounded model of the observation error. Simulation and experiments on real data shows the usefulness and the efficiency of the proposed approach.

2. Introduction

The Extended Kalman Filter (EKF) is the most popular approach used in sensor fusion for nonlinear systems [1]. This approach is based on applying the Kalman Filter (KF) algorithm on the linearization of the eventually non-linear state and measurement functions of the state model by using a first-order Taylor series expansion. The state distribution, or the posterior, can be then approximated by a Gaussian random variable which is propagated analytically through the first-order linearization of the nonlinear system.

Recently, particle filter methods have been the focus of increased interest in the field of localization problems [1,2,3,5,7]. A particle filter is a sequential Monte Carlo Bayesian estimator which is expected to provide more valuable information of the posterior especially if it has a multimodal shape or if the distributions of the noises are non Gaussian.

Nevertheless, Particle Filter methods suffer from some drawbacks. In fact, in order to explore a significant part of the state the space, the number of particles should be very large which induces complexity problems not adapted to a real-time implementation. In addition, these methods are very sensitive to non consistent measures or high measurement errors.

Several works try to combine approaches in order to overcome these shortcomings (see for example [7] and references therein). In this paper, we present a particle filter strategy which deal with interval data to solve the problem of localization of a mobile robot using dead reckoning and absolute sensors. The idea of this work becomes from two possible understandings or interpretations of an interval in one dimension:

1- An interval represents infinity of particles continuously distributed on the interval.

2- An interval represents a particle imprecisely located in the interval.

In *n* dimension, the state of a particle is described by a box.

3. Bayesian filtering for localization

Given measurements $Y_k = \{y_i\}_{i=1,\cdots,k}$, a process model and an initial guess $p(x_0)$, the goal of estimation is to determine the current state x_k. Usually Y_k is provided by exteroceptive or absolute sensors like telemeters or goniometers. The process model may be expressed as follows:

$$\begin{cases} x_{k+1} = f\left(x_k, u_k\right) + \alpha_k \\ y_k = g\left(x_k\right) + \beta_k \end{cases} \tag{1}$$

where f and g are two non-linear functions defining the state space model, x_k and y_k are the state and the observation at instant k where $x_k \in \mathrm{IR}^n$ and $y_k \in \mathrm{IR}^p$.

Vectors α_k and β_k correspond to additive noises over the state and the measurement with probability densities p_{α_k} and p_{β_k} respectively. The parameter u_k is the input of the system which is often measured using proprioceptive sensors like inertial sensors.

Within the Bayesian framework, the relevant information about $X_k = \{x_i\}_{i=1,\cdots,k}$ conditioned on all measurements $Y_k = \{y_i\}_{i=1,\cdots,k}$ consists in the posterior distribution $p(X_k|Y_k)$. In a localization process, the objective is to estimate recursively in time one of its marginals, which is the filtering density $p(x_k|Y_k)$.

The Bayesian solution to problem (1) is given by [1,2]:

$$p(x_k|Y_k) \approx p(y_k|x_k)\,p(x_k|Y_{k-1}) \quad (2)$$

$$p(x_{k+1}|Y_k) = \int_{\Re^n} p(x_{k+1}|x_k)p(x_k|Y_k)dx_k \quad (3)$$

The recursion has to be initialized with $p(x_0|Y_{-1}) = p(x_0)$, where $p(x_0)$ is some representation of a prior knowledge, e.g. a uniform distribution over some region of the state space. Equation (2) is known as the measurement update and Equation (3) as the time update.

4. Box particle Filter (BPF)

In practice, it is often hard or even impossible to solve Equations (2) and (3) analytically. An optimal solution is given by the Kalman Filter when the system is linear with Gaussian Noises. In general, when the measurement equations are non-linear or the noise distributions are non-Gausian, some form of numerical approximation need to be considered. Methods of special interest are the sequential Monte Carlo methods, or particle filters. Particle filters use an *importance sampling* approach to implement the Bayes filter to calculate (2) and (3) [3].

A contribution of this paper is to present a box particle filter which consists in handling boxes states and observations and to use constraints satisfaction techniques.

A real interval is defined as a closed and connected subset of *IR*, and a box [*x*] of IR^n as a Cartesian product of *n* intervals $[\mathrm{x}] = [\mathrm{x}(1)] \times [\mathrm{x}(2)] \times \cdots \times [\mathrm{x}(\mathrm{n})]$.

The main idea is that, at time *k*, the state space part under interest is split in N particle boxes $\left\{\left[x_0^{(i)}\right]\right\}_{i=1,\cdots,N}$ instead of “point particles” like in the original particle filter algorithm. Interval operations [4, 6, 8] are used in order to propagate each box particle and to update weight for each of them. Constraints satisfaction techniques [4, 6] are used to contract consistent particles.

Since a box particle is a box $\left[\mathrm{x}^{(\mathrm{i})}\right]$ with an associated scalar weight $\mathrm{w}^{(\mathrm{i})}$, the box particle filter (BPF) algorithm is described after:

1. Initialization:

   Set k=0, and for $i = 1,\cdots,N$ generate N boxes $\left\{\left[x_0^{(i)}\right]\right\}_{i=1,\cdots,N}$ with empty intersection in order to explore the state space part under consideration and set the weights $w_0^{(i)} = 1/N$

2. Propagation:

   For $i = 1,\cdots,N$, predict new boxes using different realizations of the noise by applying the evolution model (1):
   $\left[x_{k+1}^{(i)}\right] = f\left(\left[x_k^{(i)}\right]\right) + \left[\alpha_k^{(i)}\right]$.

3. Verification of the consistency:

   For $i = 1,\cdots,N$,

   Predict measurements $\left[z_{k+1}^{(i)}\right]$ from each box particle, $\left[z_{k+1}^{(i)}\right] = g\left(\left[x_{k+1}^{(i)}\right]\right) + \left[\beta_{k+1}^{(i)}\right]$

   Calculate the intersection with the measure: $\left[r_{k+1}^{(i)}(j)\right] = \left[z_{k+1}^{(i)}(j)\right] \cap \left[y_{k+1}(j)\right]$ for $j = 1\cdots p$, where p is the dimension of the measure.

   Calculate $A(j) = \dfrac{\left|r_{k+1}^{(i)}(j)\right|}{\left|z_{k+1}^{(i)}(j)\right|}$ for $j = 1\cdots p$, where $|X|$ is the width of $[X]$.

   If the innovation $\left[r_{k+1}^{(i)}\right]$ is not empty

   Then contract the box particle $\left[x_{k+1}^{(i)}\right]$ using the intersection box $\left[r_{k+1}^{(i)}\right]$ and forward and backward propagation (see [4])

   Else $\left[x_{k+1}^{(i)}\right]^{new} = \left[x_{k+1}^{(i)}\right]$ (the box particle stays unchanged).

4. Update of the weight of each box particle:

   For $i = 1,\cdots,N$, $w_{k+1}^{(i)} = \prod_{j=1}^{n} A(j) w_k^{(i)}$

5. Resample the particles. This eliminates particles whose weight is null.

6. Normalization: $\tilde{w}_{k+1}^{(i)} = \dfrac{w_{k+1}^{(i)}}{\sum_{j=1}^{N} w_{k+1}^{(j)}}$

7. Set $\left[x_{k+1}^{(i)}\right] = \left[x_{k+1}^{(i)}\right]^{new}$, $w_{k+1}^{(i)} = \tilde{w}_{k+1}^{(i)}$

8. Estimation:
   A minimum mean squared estimate strategy applied on the middle of the boxes intervals gives estimates of the state and of the error correvariance matrix

   $\hat{x}_k = \sum_{i=1}^{N} w_k^{(i)} x_k^{(i)}$ and $P_k = \sum_{i=1}^{N} w_k^{(i)} \left(x_k^{(i)} - \hat{x}_k\right)\left(x_k^{(i)} - \hat{x}_k\right)^T$.

9. $k = k + 1$ and return to 2.

We will discuss in the next section how to choose the width of each interval of the box.

When using the box particle algorithm, similarly problems to those of the original particle filter algorithm may occur in practice. In fact, after some iterations, the importance weights may become very skewed. Furthermore, if an aberrant measure occurs, few particles may be likely, whereas others will be degenerated and have weights close to zero, see figure (2). This problem may occur for example in the case of GPS observations when being in urban environments. A solution to this problem consists in resampling from the existing particles according to their importance weights in order to obtain independent and identically distributed samples [3,5]. It's obvious that samples with high weights are more likely to survive and the new resulting samples are dependent since they are resulting from perhaps very few ancestors. To decrease the dependency, one can add some artificial noise to the particles. A collection of resampling strategies can be found in [1].

In some case, resampling cannot answer efficiently the problem of aberrant measurements. We suggest here to use a new update of the importance weights. This new update gives less confidence in the new measure since it replaces the fourth step in the BPF algorithm by:

$$w_{k+1}^{(i)} = \prod_{j=1}^{n} A(j) w_k^{(i)} + \gamma . w_k^{(i)} \text{ with } 0 < \gamma < 1$$

The parameter γ reflects the influence ratio of new observation to learn the weights. In other words, a new measure does not affect totally the weights, but instead, it tries to keep confidence in the previous weights estimation. Fortunately, this strategy to update weights does not answer only the problem of aberrant observations and particle degeneracy, but it leads to more stability in the particle filter algorithm since the dependency between samples is decreased. Furthermore, this idea overcomes problems of computational cost of the particle filter algorithm.

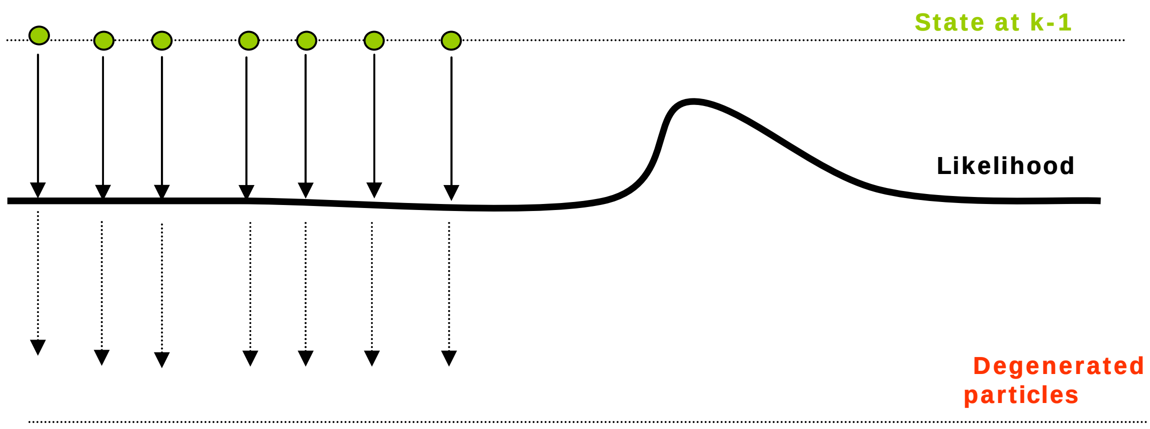

Fig.1. In the case of "point particles", and under probability frameworks, the figure shows the problem of degeneracy in the case of aberrant observation.

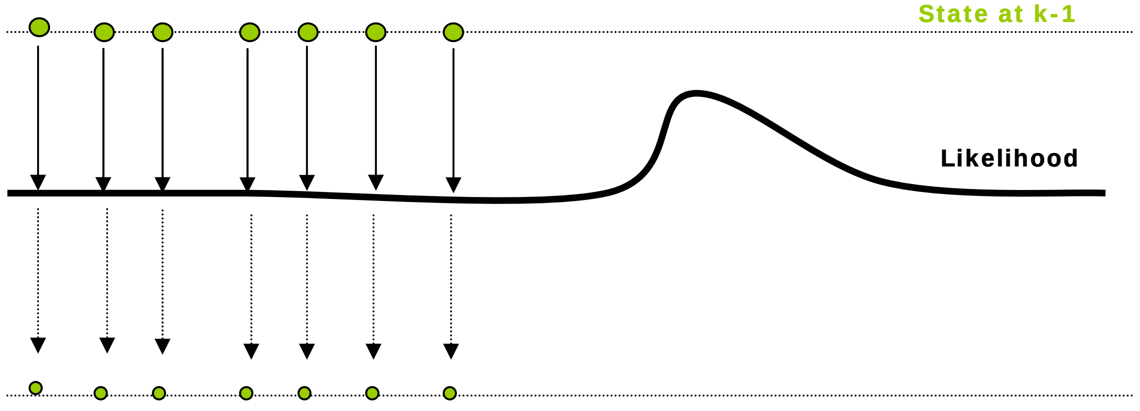

Fig.2. This figure shows that the new update $w_{k+1}^{(i)} = \prod_{j=1}^{n} A(j) w_k^{(i)} + \gamma . w_k^{(i)}$ may solve the problem of degeneracy of the particles.

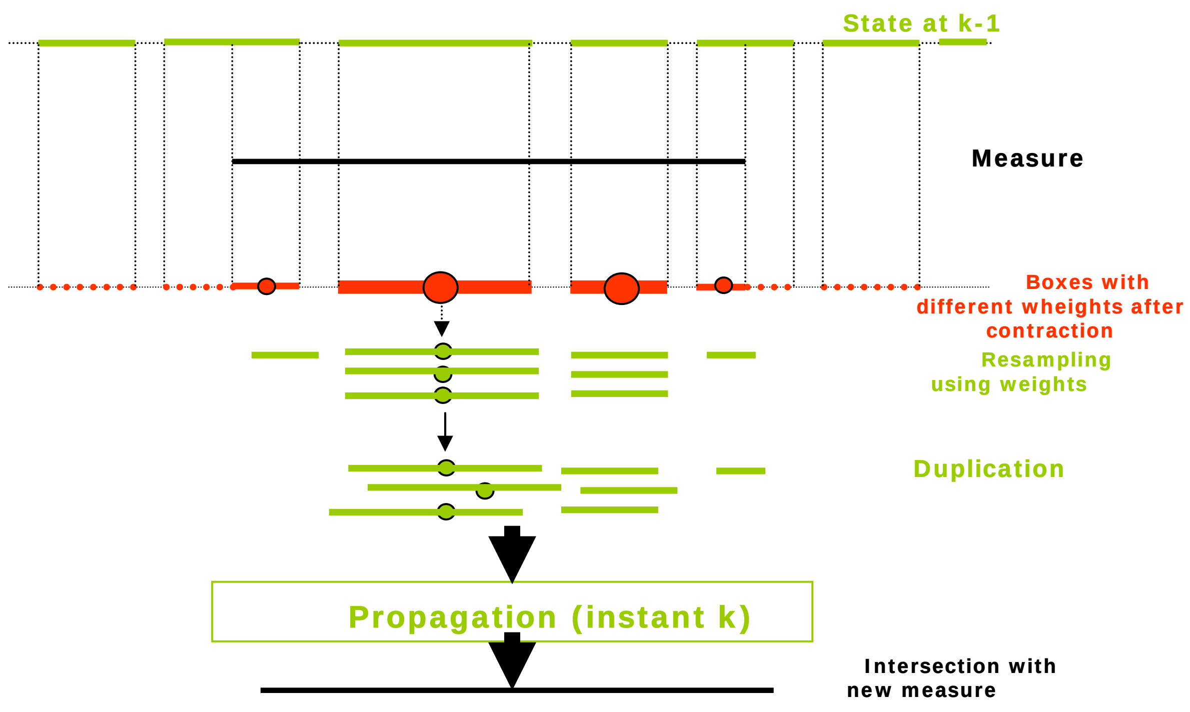

Fig.3. The figure shows the box particle filter algorithm with a resampling step.

5. Application to dynamic localization using GPS, a gyro and an odometer

The mobile frame attached to a car is represented in figure (4). The position and heading angle of the vehicle which is at time *k*, for the sake of simplicity, $[\mathrm{x}_\mathrm{k}]=[\mathrm{x}_\mathrm{k}(1)]\times[\mathrm{x}_\mathrm{k}(2)]\times[\mathrm{x}_\mathrm{k}(3)]=[\mathrm{x}_\mathrm{k}(1)]\times[\mathrm{x}_\mathrm{k}(2)]\times[\theta_\mathrm{k}]$ are calculated in time by using linear and angular velocities by the following discrete representation:

$$[x_{k+1}]=f([x_k])+[\alpha_k]=\begin{cases}[x_{k+1}(1)] & = & [x_k(1)]+[\delta_{s,k}]\cos([\theta_k]+[\delta_{\theta,k}/2])+[\alpha_k(1)]\\ [x_{k+1}(2)] & = & [x_k(2)]+[\delta_{s,k}]\sin([\theta_k]+[\delta_{\theta,k}/2])+[\alpha_k(2)]\\ [\theta_{k+1}] & = & [\theta_k]+[\delta_{\theta,k}]+[\alpha_k(3)]\end{cases} \quad (4)$$

where $\delta_{s,k}$ and $\delta_{\theta,k}$ represent respectively the elementary displacement and rotation covered by M and calculated using odometer sensors [4]. The quantities $\delta_{s,k}$ and $\delta_{\theta,k}$ are used as inputs of the system. Note here that the width of each interval should guarantee maximum variation of the variables between two instants.

The measurement of the position at time k consists here in a Global Position System (GPS) solution which is $[y_k]=[y_k(1)]\times[y_k(2)]$ after a projection onto the local frame. The width of this box can be quantified using the standard deviation estimated in real time, by the GPS receiver (GST frame).

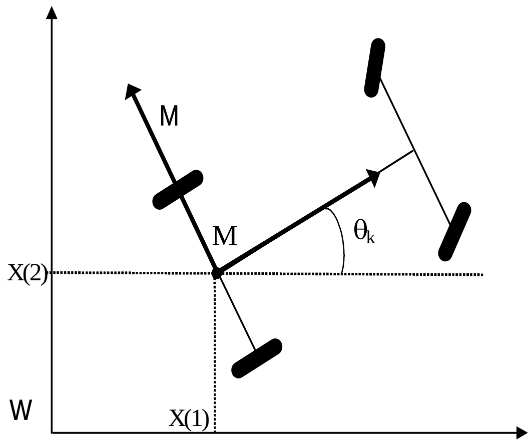

Fig.4 Mobile frame attached to the mobile robots.

The following figures illustrate the behavior of the algorithm in simulation which allows to compute estimation errors since the real poses are known.

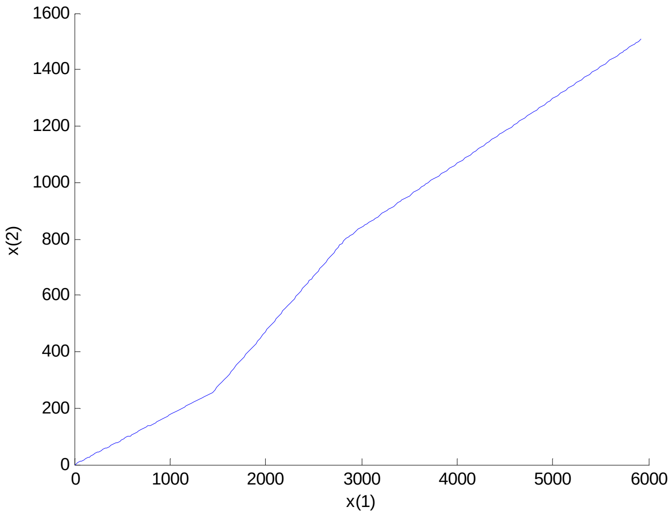

Fig. 5 The path followed by the mobile robot

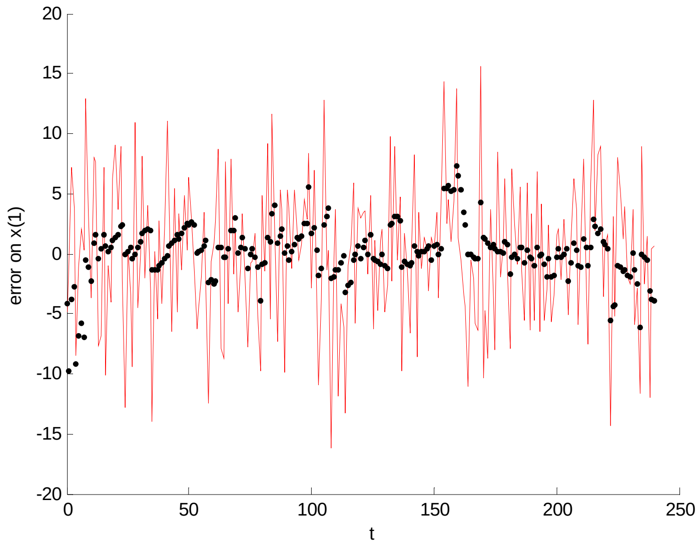

Fig.6. Points are Errors on x(1) as a function of time. And continuous line is the error of GPS measure.

Figure (6) shows a simulated path followed by a mobile robot with a variable speed. Only 64 box particle was used. Figure (7) and (8) describe the position result errors on *x(1)* and *x(2)* as a function of time and compared to the initial GPS error on each measure. More details will be given in the final paper.

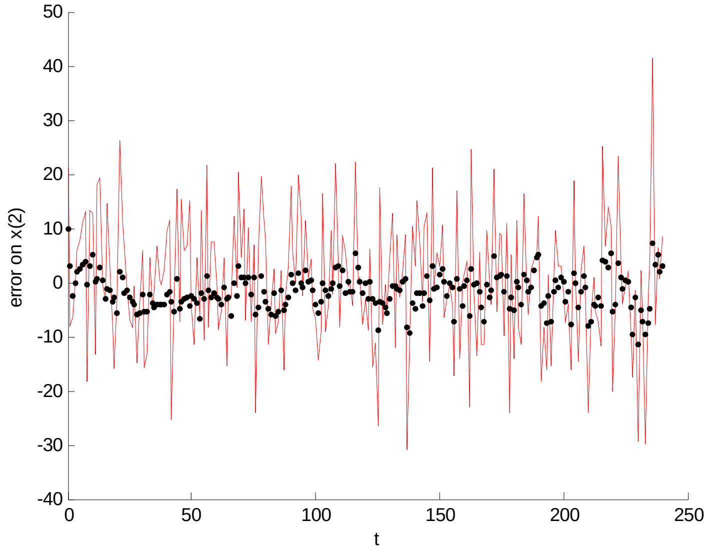

Fig.7. Points are Errors on x(2) as a function of time. And continuous line is the error of GPS measure.

6. Conclusion

A new algorithm for localization based simultaneously on particle filters and interval data has been proposed. The main idea is to try to use the interval framework which seems to be a good methodology to deal with non-white and biased measurements. Constraint Satisfaction techniques are very useful for the correction stage since a key issue in particle filtering is to find efficient methods applied to each particle. A modified update of the weights has been proposed in this paper. This update gives more stability to the filter and seems to be a good solution to the localization in urban environments since the original particle filter algorithm fails to process under similar conditions.

The results indicated that the method is able to filter effectively noisy data using only several particles. Future works will compare the method to other exiting algorithms on the basis of real experiments.

7. References

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and 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, 2002.

[2] Wen-Shiang Chen. Bayesian estimation by sequential Monte Carlo sampling. Dissertation, The Ohio State University, 2004.

[3] A. Doucet, S. Godsill, and C. Andrieu. *On sequential Monte Carlo sampling methods for Bayesian fltering*, Statistics and computing, 10:197-208, 2000.

[4] A. Gning and Ph. Bonnifait. “Dynamic Vehicle Localization using Constraints Propagation Techniques on Intervals. A comparison with Kalman Filtering”. IEEE International Conference on Robotics and Automation (ICRA 05). Barcelona, April 2005.

[5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P. Nordlund. Particle filters for positioning, navigation, and tracking. IEEE Transactions on Signal Processing, vol. 50, pp. 425- 435, Feb. 2002.

[6] L. Jaulin. “Nonlinear bounded-error state estimation of continuous-time systems”. *Automatica*, 38, 1079-1082, 2002.

[7] R. van der Merwe, J. F. G. de Freitas, A. Doucet, and E. A. Wan. The Unscented Particle Filter. Technical report, Dept. of Engineering, University of Cambridge, 2000

[8] D. Waltz. “Generating semantic descriptions from drawings of scenes with shadows”, *The psychology of computer Vision*, New York, NY pp. 19-91, 1975.