# Propagating Multiple Hypothesis for Unknown Environment Learning

**Maria-João Rendas**[*]

I3S, CNRS URA 1376
250 rue Albert Einstein, Les Lucioles 1
06560 Valbonne
France

**Stefan Rolfes**[**]

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1096 Lisboa codex
Portugal

**Abstract.** In this paper we present a computational structure that allows a mobile robot to iteratively learn a representation (map) of an *a priori* unknown environment. We assume that the robot is equipped with odometric sensors, being thus able to position itself by deadreckoning, and with local perception sensors, that provide him a partial view of objects localized in its immediate neighborhood. The goal is to build, from a growing set of these partial observations, a global description of the workspace

## 1 Introduction

In this paper we address the problem of building a representation of the geometry of the objects present in the working space of a mobile robot. We assume that the platform is equipped of two basic kinds of sensing devices: perception sensors (in our case, distance sensors) of limited range, and odometry sensors (in the studies presented here, counters of wheel turns). In figure 1 we illustrate the problem, as the estimation of the global characteristics of the environment form a set of partial views.

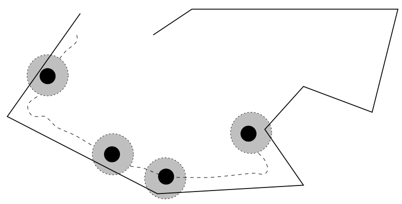


Fig. 1.: Map building from partial views.

We start by motivating our approach, in section 2 by analysing the structure of the Maximum-Likelihood (ML) estimator of the environment. In the light of the ML processor, our algorithm can be understood as a computational structure that identifies a set of plausible object models (which have a non-zero likelihood given the observed data). In section 3, we present in a general setting our approach, based on the use of fuzzy logic : we start by presenting our definition of fuzzy geometric object models, and introduce a measure of resemblance between object models that is central to our computations; we discuss after how this similarity measure is used to trigger the generation of new objects, and the recombination of distinct but similar ones; finally, we formulate the iterative updating of each object model by new data as an optimization problem. In section 4 we detail all the previous definitions for the case of fuzzy lines and circles. Finally, in section 5 we present simulation and experimental results, illustrating the performance of the algorithm.

[*] Email: rendas@estrela.unice.fr
[**] Email: rolfes@isr.ist.utl.pt

## 2 Environment Reconstruction

In this section, we briefly present results concerning the structure of the Maximum Likelihood estimation of environment geometry from perceptual and odometric data, and relate the algorithm proposed in this paper to this optimal processor. We only present notation and the basic results relevant to our discussion, referring the interested reader to [5].

We consider movement of the robotic platform in the 2D plane, denoting the position of the robot by its $(x, y)$ coordinates: $(x(k), y(k))$. We reserve the notation $X(k)$ for the *pose* and velocity of the robot, that gathers its 2D position, its orientation $\theta$ with respect to the $x$ axis, and its absolute velocity $v(k)$:

$$X(k) = [x(k)\ y(k)\ \theta(k)\ v(k)]^T .$$

We assume that $z(k)$, the *perceptual data* observed at time $k$ follows a model of known dependency on the robot pose at the corresponding instant:

$$z(k) = P_{X(k)}[\mathcal{E}] + \epsilon_z(k),$$

where $\epsilon_z(k)$ is a vector of error measurements, and $P_{X(k)}[\cdot]$ denotes the perceptual operator, dependent on the pose of the robot. If the noise density is known, this model determines the conditional density $p(z(k)|X(k), \mathcal{E})$.

Denote by $d(k)$ be the odometry measurements (output of a proprioceptive observation operator $D_k[\cdot, \cdot]$):

$$d(k) = D_k[X(k), X(k-1)] + \epsilon_d(k),$$

where $\epsilon_d(k)$ is observation noise. The density of the observed noise, along with knowledge of the determines the conditional distribution that describes the odometry measurements : $p(d(k)|X(k), X(k-1))$.

Assume further that a probabilistic Markov model of the evolution of the robot's trajectory is known, in the form of the conditional density $p(X(k+1)|X(k), \mathcal{E})$, that describes the possible trajectories with pose $X(k)$ at time instant $t_k$ given a description of the environment. Usually, this density depends only on the environment's characteristics in a neighborhood of the position at time $t_k$. This model is particularly pertinent during execution of basic perceptual behaviors, such as contour following. Since the goal of learning an internal environment representation will systematically use these low-level behaviors, we consider that, during the most informative parts of the trajectory, the robot is *effectively guided by its environment*, as the previous model indicates.

We assume also that the environment can be decomposed as the union of instances of a set of pre-specified classes of objects, that restrict the evolution of the set of families of densities in the probabilistic model that describes the perceptual data. More precisely, we admit that the environment description in $P_{X(k)}[\mathcal{E}]$ must be written as

$$\mathcal{E} = \bigcup_i \mathcal{O}_i$$

where each object is an element of a set of $M$ possible object classes: $\mathcal{O}_i \in \mathcal{A}_1$ or $\mathcal{O}_i \in \mathcal{A}_2$ or $\cdots \mathcal{O}_i \in \mathcal{A}_M$. We assume no restrictions in the spatial arrangements of the objects with respect to each other.

Given the odometry and perception data, we want at each time $t_k$ to find the environment description $\hat{\mathcal{E}}$ such that the following *likelihood function* is maximized:

$$\hat{\mathcal{E}} = \arg\max_{\mathcal{E}} p(D^k, Z^k|\mathcal{E})$$

whre $D^k$ and $Z^k$ denote the set of all odometry and perceptual measures collected up to instant $t_k$. i.e., we must find the environment description that best explains the odometry and perceptual data.

Note that our goal is *not* to determine the best estimate of the robot's trajectory $X^k$. However, the close interdependency between the odometry and perceptual data will implicitly require the generation of this estimate. In fact, it can be shown that [5] the likelihood function for this problem can be written in the following form:

$$p(D^k, Z^k|\mathcal{O}_{m=1}^M) = \int \prod_{m=1}^M p(Z^m|X^m, \mathcal{O}_m)\, p(X(0)) \prod_{m=1}^M p(X_1^m|X_{n_{m-1}}^{m-1}, \mathcal{O}_m) \prod_{i=2}^{n_m} p(X_i^m|X_{i-1}^m, \mathcal{O}_m) p(D^k|X^k) dX^k$$

In the previous equation :

- $Z^m, D^m$ are subsets of $Z^k$ and $D^k$, respectively, that group all observations related to object $\mathcal{O}_m$;
- $X^m$ is the subset of robot's poses during which object $\mathcal{O}_m$ has been observed: $X_k \in X^m \leftrightarrow z_k \in Z^m$.

This equation shows that that link between the several objects is done by conditional density of the odometric data, and by the density of the complete trajectory, that introduces dependencies between the points correspondent to adjacent objects. However, a special form of recursivity can be detected in the previous expression. It can be shown that the likelihood function can be written for a set of observations corresponding to the same object (we omit the dependency on the object index $m$ in the following expression) as

$$p(Z^{k+1}, D^{k+1}|\mathcal{O}) = \int dX(0)p(X(0)) \int dX_{k+1}p(X_{k+1}, Z^{k+1}, D^{k+1}|\mathcal{O}, X(0))$$

where the density in the inner integral is recursively calculated as:

$$p(X_{k+1}, Z^{k+1}, D^{k+1}|\mathcal{O}, X(0)) = \int p(X_k, Z^k, D^k|\mathcal{O}, X(0))p(X_{k+1}|X_k, \mathcal{O})\, p(z_{k+1}|X_{k+1}, \mathcal{O})\, p(d_{k+1}|X_{k+1}, X_k)\, dX_k$$

It can also be seen that for the complete chain, we obtain

$$L = \int \ldots \int \int p(X(0))p(X^1_{n_1}, Z^1, D^1|\mathcal{O}_1, X(0))...p(X^M_{n_M}, Z^M, D^M|\mathcal{O}_{M-1}, X^{M-1}_{n_{M-1}}).dX^M_{n_M} \ldots dX^1_{n_1}dX(0) \tag{1}$$

The previous expression of the likelihood function shows that the information that positioning and map building (i.e., determination of the robot's position and identification of the objects $\mathcal{O}_m$) cannot be done independently, i.e., both problems are tightly coupled when the robot is executing a perception based behavior.

## 3 Fuzzy Approach to Object Identification

The goal of the structure presented in this paper is to identify, from the set of odometric and perceptual measurements, a set of plausible hypotheses for the objects present in the workspace of the robot (those that must be considered to compute the likelihood function presented in the previous section). Each hypothesis that is identified by our algorithm defines an object $\lambda$ with a "fuzzy geometry," corresponding to a fuzzy subset of crisp elements of each admissible object class $\mathcal{A}$. In this way, by grouping in the same fuzzy identity a subset of the possible instances of each object type, we are able to limit our analysis to a finite number of cases, being able to manage the high computational complexity of the problem.

**Fuzzy object models** We assume that the geometry of the crisp objects that belong to a given class $|calA_n$ is described by a vector of parameters $v(\cdot) \in \mathcal{V}_n$, which defines it as the set of points $(x, y)$ that belong to the object as those that verify some parametric equation of general form:

$$C_n(x, y, v) = 0, \qquad (x, y) \in \mathbf{R}^2 \qquad v \in \mathcal{V}_n.$$

We define a "fuzzy object" $\lambda$ as a fuzzy subset of the parameter space of each object class, $\mathcal{V}_n$, i.e., we consider the elements of the parameter vector that specifies each class as fuzzy numbers. The fuzzy prototype $\lambda$ is thus defined by a membership function defined over $\mathcal{V}$:

$$\begin{aligned} \lambda : \mathcal{V} &\rightarrow [0, 1] \\ v &\rightsquigarrow \mu_v(\lambda) \end{aligned}$$

This fuzzy object model determines a fuzzy subset of points of the 2D space by the following corresponding membership function (defined over $\mathbf{R}^2$):

$$\mu_v(x, y) = \max_{v \in V^*} \mu_\lambda(v) \qquad\qquad V^* = \left\{ v^* | v^* = \arg\min_{v \in \mathcal{V}_n} C_n(x, y, v) \right\}$$

i.e., we consider the degree of membership of a given point in the fuzzy prototype $\lambda$ as the membership of the parameter vector that best fits the data point $(x, y)$. In the next section we illustrate this definition for the case of fuzzy lines and fuzzy circles.

**Update of fuzzy prototypes**

We consider in this paragraph the problem of updating each prototype as new data is collected by the robot. We assume that the membership functions $\mu_\lambda(v)$ have (for each class of objects) a fixed shape, only differing in their locations:

$$\mu_\lambda(v) = \mu_0(v - v^*(\lambda)).$$

where $v^*(\lambda)$ is the "center" of the fuzzy prototype $\lambda$. For each new measurement $z_k$, we update of $\mu_\lambda(v))$ by minimizing the following quadratic criterion:

$$\mathcal{F}^k_\lambda(v) = \sum_{i=1}^{k} \mu^i_\lambda(z_i) C^2(z_i, v),$$

where each association factor $\mu^i_\lambda(z_i)$ is computed using the value of $v^*(\lambda)$ at time instant $i-1$:

$$\mu^i_\lambda(z_i) = \mu_0\left(v - v^*_{i-1}(\lambda)\right).$$

The preceding equation shows that the optimization of $\mathcal{F}^k_\lambda(v)$ does not lead to the optimal solution (in classical mean-square sense), as it is proposed, for instance, in [1]. However, this form of the optimization criterion is necessary to lead to an iterative solution, i.e., one that does not require memorization of all previously received data points.

**Creation and recombination of prototypes**

Data points with a weak degree of membership with respect to all existing prototypes may correspond either to outliers or to detection of new objects. For each such data point, a set of new prototypes is created, by sampling the subset of $\mathcal{V}$ such that $\mu_\lambda(z) \approx 1$. Creation of new prototypes corresponds (logically) to the generation of a new set of plausible hypotheses concerning the characteristics of the object that has been detected. If a new object has effectively been detected, future measures will be associated to a certain number of these new prototypes, which, by minimization of $\mathcal{F}^k_\lambda(v)$ will converge towards a correct description of the new object. Wrong hypothesis will present a decreasing degree of association, and, as we will see next, will be eliminated by recombining them with other (stronger) prototypes.

Since, for each new measure, several prototypes can be updated, initially distinct hypothesis can become increasingly similar, making the set of represented object non-minimal. This redundancy unnecessarily increases the computational complexity of the algorithm. To keep this redundancy to a minimum, we define a a non-symmetrical binary similarity relation $D(\lambda_1.\lambda_2)$ that is, by definition, the (fuzzy) truth value of the following proposition [3]: $D(\lambda_1, \lambda_2)$: "prototype $\lambda_1$ is a subset of $\lambda_2$", i.e.,

$$\mu_D(\lambda_1, \lambda_2) = \lfloor \lambda_1 \subset \lambda_2 \rfloor = \mu_D(\lambda_1, \lambda_2) = \frac{\int \mu_{\lambda_1 \wedge \lambda_2}}{\int \mu_{\lambda_2}}.$$

and where all integrals are computed over the entire 2D plane. The non-symmetric relation $D(\cdot, \cdot)$ is a normalized version of the symmetric binary relation $\mu_S(\cdot, \cdot)$

$$\mu_D(\lambda_1, \lambda_2) = \frac{\mu_S(\lambda_1, \lambda_2)}{int \mu_{\lambda_2}} \qquad \mu_S(\lambda_1, \lambda_2) = \int \mu_{\lambda_1 \wedge \lambda_2} \tag{2}$$

which can be recursively updated, for each new observation $z_k$, by

$$\mu_S(\lambda_1, \lambda_2)_k = \mu_S(\lambda_1, \lambda_2)_{k-1} + \min\left(\mu_{\lambda_1}(z_k), \mu_{\lambda_2}(z_k)\right).$$

When a prototype $\lambda_1$ becomes a subset of another prototype $\lambda_2$, they are recombined to form a new prototype $\lambda^*$. This recombination implies the definition of the membership function of the new prototype $\mu_{\lambda^*}$ and the determination of its similarity with respect to all other prototypes $\mu_S(\lambda^*, \lambda)$. We define the resulting fuzzy object $\lambda^*$ as the union of the two original fuzzy sets:

$$\lambda^* = \lambda_1 \bigcup \lambda_2, \qquad \mu_{\lambda^*}(x, y) = \max\left(\mu_{\lambda_1}(x, y), \mu_{\lambda_2}(x, y)\right).$$

The relation $\mu_S(\lambda^*, \lambda)$ is, by definition

$$\mu_S(\lambda^*, \lambda) = \int \mu_{\lambda^* \wedge \lambda} = \mu_S(\lambda_1, \lambda) + \mu_S(\lambda_2, \lambda) - \int (\mu_{(\lambda_1 \cap \lambda) \cup (\lambda_2 \cap \lambda)}. \tag{3}$$

[3] we denote y $\lfloor p \rfloor$ the truth value of proposition $p$.

The last term in this equation cannot be determined from knowledge of $\mu_S(\cdot,\cdot)$. However, we can find upper and lower bounds to it. Use of a lower bound would result in an over-estimation of resemblance, which might lead to the recombination of non-redundant prototypes. For this reason, we decided to use a lower bound to approximate the previous expression, using the following upper bound to the integral in (3):

$$\int (\mu_(\lambda_1 \cap \lambda) \cup (\lambda_2 \cap \lambda) \leq \min \left( \int \mu_{\lambda_1 \cap \lambda}, \int \mu_{\lambda_2 \cap \lambda} \right) \leftrightarrow \mu_S(\lambda^*, \lambda) \geq \max(\mu_S(\lambda_1, \lambda), \mu_S(\lambda_2, \lambda)) .$$

**Algorithm Structure** We can now present the structure of our algorithm:

$\mathcal{E} = \emptyset$
for *each observation* $(z_k)$
  $\forall \lambda \in \mathcal{E}$, *compute* $\mu_\lambda(z_k)$
  *update* $\mu_S(\cdot,\cdot)$
  if $\mu_\lambda(z_k)$
    *update* $\lambda$,
  if $\neg(\exists \lambda : \mu_\lambda(z_k))$
    *generate new prototypes* $\eta_i$ *such that* $\mu_{\eta_i}(z_k)$
    $\mathcal{E} = \mathcal{E} \bigcup_i \eta_i$
  if $\mu_D(\lambda_1, \lambda_2)$
    $\lambda^* = \lambda_1 \bigcup \lambda_2$,
    *compute* $\mu_S(\lambda^*, \lambda)$

where $\neg x$ represents the logical complement of $x$.

In the next section we specialize this algorithm to geometric environment description using fuzzy lines and circles.

## 4 Environment Representation using Fuzzy Lines and Fuzzy Circles

**Fuzzy lines and circles** A "crisp" line is uniquely represented by its direction $a$ and its coordinate at the origin $b$. The parameter space for this class of objects is thus two-dimensional $\mathcal{V} = \mathbf{R}^2$: $v = (a, b)$. The parametric equation that defines the set of points that belong to the line is

$$C(x, y, v) = 0 \Leftrightarrow y - ax - b = 0$$

Our definition of fuzzy line $\lambda$ considers that $b$ is a fuzzy number defined by the membership function $\mu_{b(\lambda)}(b)$:

$$\mu_{b(\lambda)} : \begin{array}{rl} \mathbf{R} \rightarrow & [0,1] \\ b \rightsquigarrow & \mu_{b(\lambda)}(b) \end{array} .$$

The membership function that defines the fuzzy line as a subset of $\mathcal{V}$ is thus :$\mu_\lambda(a, b) = \delta(a - a(\lambda))\mu_{b(\lambda)}(b)$. Its membership function as a fuzzy subset of the 2D space is easily obtained:

$$\mu_\lambda(x, y) = \max_b \mu_{b(\lambda)}(y - ax).$$

A "crisp" circle is defined by its radius $r$, and by its center $(x_c, y_c)$. For this class, we see thus that $\mathcal{V} = \mathbf{R}^+ \times \mathbf{R}^2$. Our definition of fuzzy circle corresponds to taking $r$ as a fuzzy number, defined by $\mu_{r(\lambda)}(r)$, and thus :

$$\mu_\lambda(r, x_c, y_c) = \delta(x - x_c(\lambda))\delta(y - y_c(\lambda))\mu_{r(\lambda)}(r)$$

As a subset of $\mathbf{R}^2$,

$$\mu_\lambda(x, y) = \max_r \mu_{r(\lambda)} \left( \sqrt{(x - x_c(\lambda))^2 - (y - y_c(\lambda))^2} \right) .$$

The preceding equation is derived from the parametric equation that defines the points that belong to the circle:

$$X(x, y, r, x_c, y_c) = (x - x_c(\lambda))^2 - (y - y_c(\lambda))^2 = r^2 .$$

**Update Equations** We rewrite here the quadratic criterion defined previously:

$$\mathcal{F}_\lambda^k = \sum_{i=1}^{k} \mu_\lambda^i(z_i) C^2(z_i, \lambda) .$$

where $z_i = (x_i, y_i)$ and $C(\cdot, \cdot)$ has been defined above both for lines and for circles. At each iteration (arrival of a new data point), a new pair $(a(\lambda), b(\lambda))$ must be associated to each line and a new triple $(r(\lambda), x_c(\lambda), y_c(\lambda))$ to each circle. It can be shown that [4] the solution to these optimization problems are

$$\begin{bmatrix} \hat{a}(\lambda) \\ \hat{b}(\lambda) \end{bmatrix} = A^{-1} D \qquad\qquad \begin{bmatrix} \hat{x}_c(\lambda) \\ \hat{y}_c(\lambda) \\ \hat{r}(\lambda) \end{bmatrix} = C^{-1} E$$

where the elements of matrices $A$ $(2 \times 2)$ and $C$ $(3 \times 3)$ and of vectors $D$ and $E$ are functions of the (fuzzy) moments of the data, of general definition

$$\mu_{pq}^k = \sum_{i}^{k} \mu_\lambda^i(z_i) x_i^p y_i^q , \qquad \mu_{pq}^k = \mu_{pq}^{k-1} + \mu_\lambda^k(z_k) x_k^p y_k^q .$$

We conclude thus that the recursive update of the prototypes using criterion $\mathcal{F}_\lambda^k$ implies the propagation (along time) of the fuzzy moments.

**Recombination** Whenever two prototypes are recombined, the moments associated to the new prototype must be determined. We present here the equations that define the moments of the new prototype that results of the recombination of $\lambda_1$ and $\lambda_2$ :

$$\lambda^* = \lambda_1 \cup \lambda_2$$

$$m_{ij}^\lambda = \int_{\Re^2} \mu_\lambda(x, y) x^i y^j \, dx dy$$

and using the fact that if $\lambda = \lambda_1 \cup \lambda_2$ then $\mu_\lambda(x, y) = \max(\mu_{\lambda_1}(x, y), \mu_{\lambda_2}(x, y))$, we may write, denoting $\mu_{\lambda_i}$ by $\mu_i$ and omitting dependency on $(x, y)$

$$m_{ij}^\lambda = \int_{\mu_1 > \mu_2} \mu_1 x^i y^j + \int_{\mu_2 > \mu_2} \mu_2 x^i y^j = m_{ij}^1 + m_{ij}^2 - \int \min(\mu_1, \mu_2) x^i y^j . \tag{4}$$

The last term of the right hand-side of the previous equation can be approximated using the inclusion relation $\mu_D(\cdot, \cdot)$ :

$$\int \min(\mu_1, \mu_2) x^i y^j = \int \mu_{1 \wedge 2} x^i y^j \simeq \int \frac{\int \mu_{1 \wedge 2}}{\int \mu_2} \mu_2 x^i y^j = \mu_D(\lambda_1, \lambda_2) m_{ij}^2 \tag{5}$$

where we used the fact that, for similar prototypes,

$$\mu_2(x, y) \mu_D(\lambda_1, \lambda_2) \simeq \min\left(\mu_{\lambda_1}(x, y), \mu_{\lambda_2}(x, y)\right) ,$$

yielding

$$m_{ij}^{\lambda^*} = \begin{cases} m_{ij}^1 + (1 - \mu_D(1,2)) m_{ij}^2 & , \quad \mu_D(1,2) \geq \mu_D(2,1) \\ m_{ij}^2 + (1 - \mu_D(2,1)) m_{ij}^1 & , \quad \mu_D(2,1) > \mu_D(1,2) \end{cases} \tag{6}$$

## 5 Presentation of the results

We tested the algorithm with : (i) simulated data, and, (ii) measures taken in a real experiment.
**Simulated data** : We defined a chain of arcs and line segments and randomly generated points along these arcs and line segments. The two simulated environments are shown in figure 2.

Robot's movement has been simulated by creating points at an approximately constant rate, corresponding to the robot's speed, and introduced a gaussian perturbation to this constant rate. A random perturbation has been added to the measurements of distance to the objects, simulating the perceptual data. We also introduced outliers, occurring at fixed rates (0%, 10% and 15%), with direction uniformly distributed in $[0, 2\pi]$, and range uniformly distributed between fixed minimum and maximum distances.
**Real data** : Our experiments were done with the mini-robot Khepera, which is a robot of circular shape with a diameter of approximately 5.2cm. It is supported by two wheels (of a diameter of ca. 15mm) and two small teflon balls. Every wheel can be driven in both directions by a DC motor coupled to the wheel through a 25:1 reduction gear. An incremental counter placed at the motors yields the odometric measurements. The robot is also equipped with 8 infra-red proximity sensors, which are, in the configuration used, its only means for perceiving its environment.

We placed the robot inside a closed region delimited by objects of circular and polygonal shape. The robot was then run on a completely autonomous mode, using a simple control algorithm to keep it in a trajectory parallel to the wall. We halted the robot after the completion of an entire tour of its environment. Three of its infra-red sensors (those placed at directions 0, 45 and 90 degrees with respect to the direction of motion) are used for trajectory control. As the robot progresses, the odometric measurements are used to infer the robot trajectory and are combined with the infra-red measurement of the sensor orthogonal to the direction of motion to infer detections of points in the contour, which are processed by the algorithm.

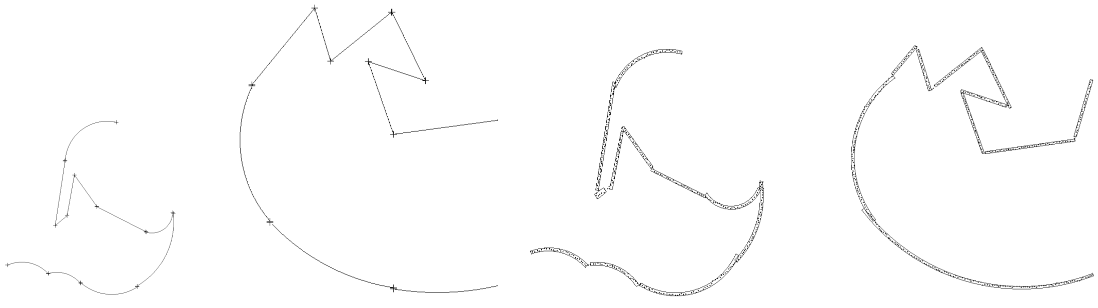
The simulated environments were designed in order to be able to evaluate the performance of the algorithm in the following situations :

- sequence of two convex cap or convex cup arcs
- Transition from a convex cup to a convex cap arc
- Transition from an arc to an arc with lower (or larger) curvature
- Transition from an arc o a line segment (or vice-versa)
- sequence of zig-zagging line segments

The membership functions of the fuzzy numbers that describe the coordinate at the origin of the lines, $b$, and the radius of the circles, $r$, are of trapezoidal shape, determining, for each prototype a non-zero measure region of maximal membership in $\mathbf{R}^2$.

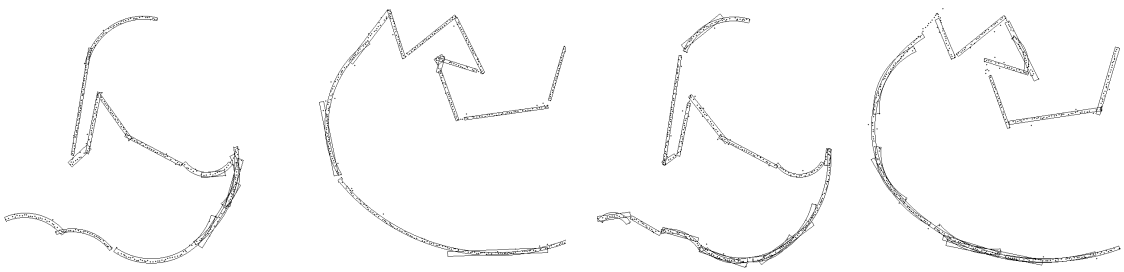
The results of the algorithm for simulated data are shown in figures 2–3. In the figures presented below, all detected objects are represented along with their region of maximal uncertainty ($\mu_\lambda(x, y) = 1$). Detections are also shown in the figures, in order to better appreciate the algorithm's behavior.

With no outliers, figure 2, the algorithm correctly identifies the constituent objects. In the presence of outliers, we see that additional redundant prototypes are created. We point out that all parameters were held constant for all three outlier rates, with the goal of analyzing the algorithm's behaviour in unmodeled error situations. Figures 4–4 show the results of the real explorations in two different closed environments.



(a) (b)

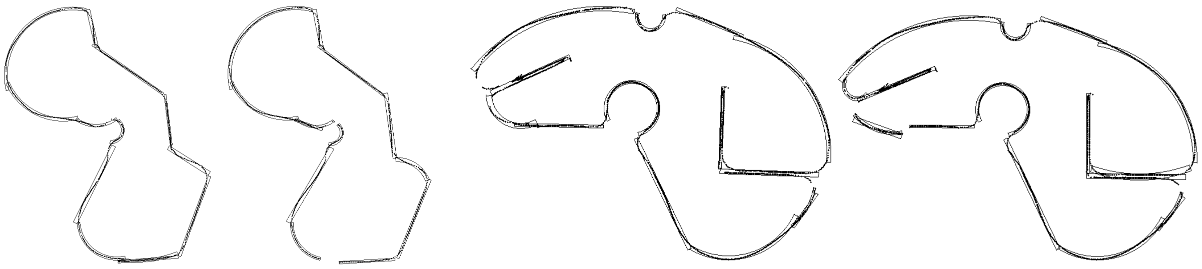
Fig. 2.: (a) Environments for simulated data; (b) Simulations : outlier rate of 0%.



(a) (b)

Fig. 3.: Simulations : outlier rates of (a) 10% and (b) 15%.

To demonstrate the stability of the algorithm, we performed two separate explorations of each contour. Our goal in these experiments was tow-folded : *(i)* verify whether a high-level description of the environment was identified by the algorithm, and whether it closely approximated the geometry of the curved environment; (ii) evaluate the stability of the algorithm, by comparing the results of the two independent explorations. The results of the algorithm for the two experiments show that we can affirmatively answer both questions : the algorithm identified a set of line segments and arcs of circle that approximates the curved shape, and the prototypes identified in both experiments, although not the same, present a high degree of resemblance. This last point is specially important if the algorithm's result is to be used for future robot positioning. However, this possibility requires that odometry errors (which are treated in a simplistic manner by the present version of our algorithm) are dealt with in a more sophisticated way, by propagating (as the Maximum Likelihood solution indicates) the uncertainty with respect to the positioning and orientation of each object. We are currently working in this direction.



(a) (b)

Fig. 4.: (a) Environment 1 : chain of polygonal objects followed by a curved contour including a small circular object. (b) Environment 2 : curved border with two curved and two polygonal internal walls.

## References


1. R. Krishnapuram, O. Nasraoui, H. Frigui. The Fuzzy C Spherical Shells Algorithm: A new Approach, Transactions on Neuronal Networks IEEE, Sept. 1992, Vol. 3, Nbr. 5
2. S. Gottwald. Fuzzy Sets and Fuzzy Logic: Foundations of Application from a Mathematical point of view, Wiebaden, Vieweg Verlag, 1993
3. Kenneth G. Manton, Max A. Woodbury and H. Dennis Tolley, *Statistical Applications using Fuzzy Sets*, Wiley series in probability and Mathematical Statistics, John Wiley & Sons, 1994.
4. Stefan Rolfes, M. J. Rendas, M.-C. Thomas, "Mobile Robots: Learning an unknown Environment," Rapport CNRS/I3S December 1995.
5. M. João Rendas, "Maximum Likelihood Environnemen Estimation," Internal report CNRS/I3S Mai 1996.


This article was processed using the TEX macro package with SIRS96 style