Incremental Construction of a Landmark-based and Topological Model of Indoor Environments by a Mobile Robot

Hanna Bulata Michel Devy
L.A.A.S. - C.N.R.S.
7 ave. du Colonel Roche, 31077 Toulouse Cedex - FRANCE
E-mail: hanna@laas.fr, michel@laas.fr

Abstract

This paper deals with the perception subsystem of a mobile robot which must navigate in a structured environment. We will consider the exploration task; the robot must build successive snapshot models from sensory data acquired from a laser range finder (LRF), and fuse them in a global model so that it can localize itself with respect to a pertinent reference frame. Structuration rules are required to limit the complexity of the modeling process. First of all, we only extract from the sensory data, useful landmarks which correspond to characteristic local feature groupings, like wall corners, doors, corridor crossings, ...; each landmark has its own frame and its own geometrical model. Then, correlated landmarks are kept together at an area level; finally, the global model is built from the relationships between area frames, providing the topological description of the world. For each level (landmark, area, environment), the model is represented by a random vector and a covariance matrix, updated through the use of an Extended Kalman Filter (EKF).

1 Introduction

A mobile robot should be able to perceive its environment before any task execution. The execution of safe movements as well as the achievement of tasks in an intelligent way depend on the quality of data *gathered* by the robot. The robot has an odometer which delivers positional information with respect to (noted w.r.t.) a specific reference frame and a laser range finder (noted LRF) which delivers informations about the world surrounding the robot. But, sensory data are *always* imprecise due to sensors limitation. Tasks that depend on such data, will fail if they cannot accommodate the errors [8].

A major source of error is introduced by the robot motion. The rough estimation of the robot position available from the odometric system may be corrupted by a large cumulative error due to unpredictable slippage of the wheels. So, a relocation procedure is required: localization is a process of "looking" for and "tracking" expected events [5]. An event is "expected" if it can be predicted from either an internal model of the environment or from previously observed events. It is possible to estimate errors on the robot position from matchings between events perceived in successive perceptions.

In our previous works, these perceptual events corresponded only to single features extracted from the sensory data: straight lines [6] or heterogeneous features [4]. The robot position was estimated w.r.t. a global reference frame. For long experiments, these methods have important drawbacks: features are numerous and ambiguous, so that the matching process between features extracted in the current perception and the ones which are already known, can be very complex, and the global map can be very cumbersome if all correlations are taken into account.

We have proposed a localization method based on the grouping of perceptual features called *landmarks* [3]. We argue that the localization could be easier when based on these landmarks. Approximative knowledges about landmark positions was given initially to the robot; landmarks correspond either to feature configurations found naturally in a structured indoor environment such as parallel walls, corners, doors, ..., or to artificial patterns added for the localization purpose. The landmark shape was supposed perfectly known. In this paper, we present the modeling and the localization procedures, required when the robot explores a large indoor environment, with several places (rooms, corridors, ...). These two processes are dependent as building the world map requires that the robot position be known accurately so that consistency could be ensured [7]. The hierarchical structure of our model is described in the section 2.

The map describing the robot environment must be built during a specific learning step or directly during the task execution. This map involves uncertain relationships between its components, i.e. between landmark and robot positions; landmarks are first located in the robot frame before being *transfered* to the current reference frame. This dependency is expressed by *correlations* between the error on the robot and the landmark positions. Therefore, when the robot position is updated, all map elements correlated with it must be updated to ensure consistency.

Kalman filtering is widely used as a tool to deal with the incremental construction of a stochastic map. The uncertain relationships between its elements, are represented by covariance matrices. At each robot pose and data acquisition, the map components and their dependencies are updated by the filter. Therefore, even if a landmark is unperceived, its position and corresponding uncertainty can be updated. The section 3 describes how to update a stochastic map, as defined in [8].

In this paper, two problems are specifically addressed. First of all, in the section 4, we describe the *landmark extraction function*, required when the robot starts without any knowledge on the environment, in order to build **landmark models**. Then in the section 5, we detail our landmark-based localization function, which updates all the knowledges of our hierarchical model (landmark models, landmark, area and robot positions) while the robot is moving. At last, we present in the section 6 experimental results which validate this approach.

2 A Hierarchical Model

In order to navigate in an unknown and large indoor environment, like buildings for example, our robot builds a *hierarchical model*. Like in [2], the knowledge is distributed on different levels.

The geometrical level: the landmark models.

Only useful feature configurations, called *landmarks*, are learnt. As in [1], we have proposed a utility function [3] related to the improvement we can obtain on the accuracy of the robot position, when these features are used for the relocation. A landmark has a description which must be accurate enough according to a proper frame, so that it can be recognized and located w.r.t. the robot frame.

The symbolic level: the area models.

Once its pose has been computed w.r.t. the robot frame, a landmark is known only by a 2D transform. Then the robot position is required in order to estimate the landmark position w.r.t. the environment. It is not computed w.r.t. the global reference frame, which can be far from the current robot position. The errors involved during the transfer of the landmark pose from the robot frame to a world reference frame must be minimized; we want to avoid "lever arm" effects which could affect the estimate of the landmark pose if the error on the robot orientation and the distance between the origins of the world and the landmark frames were important.

So, a landmark pose is known w.r.t. the local area in which the robot is currently moving; it is memorized with the positions of the other landmarks extracted in the same area.

The topological level: the environment model.

The robot explores a current area, until it crosses a door. An area corresponds to a semantical entity (a room, a corridor, ...), but especially, to perceptual constraints: when executing a task in an area $\mathcal{A}_i$, the robot can look only for the landmarks linked to $\mathcal{A}_i$. The area size can vary according to the sensor range and to the environment structure.

The figure 1 shows an environment with 3 areas; once the robot enters a new area, it must define a new local reference frame and locate itself according to this one. In our current modeling system, this area frame is anchored on the door crossed by the robot. If our robot discovers the areas $\mathcal{A}_1$, then $\mathcal{A}_2$ and at last $\mathcal{A}_3$, then local area reference frames will be laid on the doors $P0 \leftrightarrow 1$ for $\mathcal{A}_1$, $P1 \leftrightarrow 2$ for $\mathcal{A}_2$ and $P2 \leftrightarrow 3$ for $\mathcal{A}_3$. The relationships between these area frames and the global reference frame defined according to high level requirements, are kept in the environment model with the current robot position computed w.r.t. the global frame.

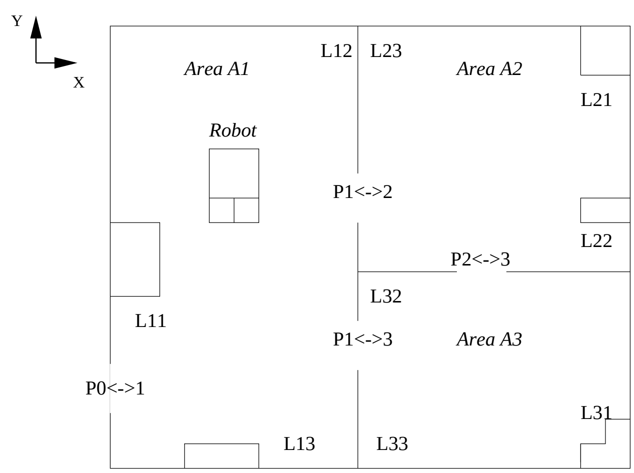

Figure 1: *An environment with 3 areas*

According to the feature-based approach, the knowledge distribution implies that on the one hand, some features -the ones which do not belong to landmarks- are forgotten, and on the other hand, that some correlations between the perceived events and the robot position, are neglected: for example, we do not kept the correlations between segments which can be seen together, but which belong to different landmarks. Our experiments have shown that the relocation results do not suffer from these model simplifications.

3 The stochastic model

Many authors [8] [6] [5] represent an incrementally built model, by a stochastic map. The modeling task involves that, from different positions, the robot can measure from its sensors, geometrical descriptions of objects and spatial relationships between objects and itself. The uncertainties on these measurements can be estimated by the propagation of the known errors given in the sensor models: the odometry model for the robot displacement, the LRF model for the sensory data.

The geometrical description of an object l is represented by a set of features; each feature i has a minimal parametric representation, computed w.r.t. a reference frame k, and given by a random vector $X^k_{l,i}$. In our model, the features can be 2D segments ($S = (pq)^t$, p, q minimal representation of the straight line which holds up the segment) or 2D points ($P = (xy)^t$).

A spatial relationship corresponds to the position of an object i w.r.t. a reference frame k; it will be also represented by a random vector, R^k_i: a translation and a rotation in a 2D map. In our model, such a vector could represent a landmark position w.r.t. the robot or an area frame, an area position w.r.t. the global frame, or the robot position w.r.t. a landmark, an area or the global frame.

Each random vector X is modeled by estimating the mean $\hat{X}$ and the covariance matrix $C(X)$. For a spatial relationship R:

$$R = \hat{R} + \epsilon_R; \quad C(R) = E[\epsilon_R {\epsilon_R}^t]$$

$$\hat{R} = \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{pmatrix} \qquad C(R) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \end{pmatrix}$$

At each level, a model is then represented by a *state vector* and by an *uncertainty state matrix*. Each element of the state vector is also a vector R, S or P: only S and P vectors for the landmark models, only R vectors for the area and environment models. The first component of the state vector is always the robot position. For a model containing n elements, the representation is :

$$V = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} \quad C(V) = \begin{pmatrix} C(X_1) & \dots & C(X_1, X_n) \\ C(X_2, X_1) & \ddots & \\ C(X_n, X_1) & \dots & C(X_n) \end{pmatrix}$$

Let us recall when such a model must be *updated.*

3.1 Updating the robot position

When the robot moves from an initial position Rr^k_i w.r.t. a reference frame k, with an uncertain relative motion $\Delta_{i,i+1}$ (figure 2-a), the new robot position, before any perceptual relocation, can be computed by the use of a transform composition (noted f):

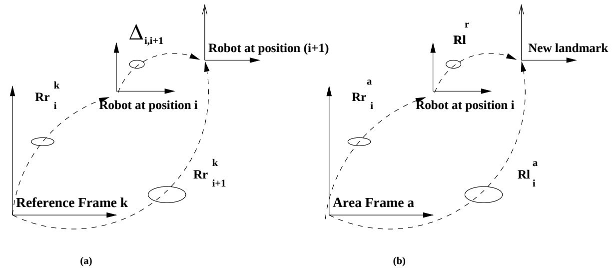

Figure 2: *Spatial relationships*

$$Rr^k_{i+1} = f(Rr^k_i, \Delta_{i,i+1}) \tag{1}$$

$\hat{Rr}^k_{i+1}$ can be estimated from the estimates $\hat{Rr}^k_i$ and $\hat{\Delta}_{i,i+1}$.By the use of a linearization of f, errors can be propagated; because Rr^k_i and $\Delta_{i,i+1}$ are independent spatial relationships, the new uncertainty on the robot position is :

$$C(Rr^k_{i+1}) = J_1 C(Rr^k_i) {J_1}^t + J_2 C(\Delta_{i,i+1}) {J_2}^t$$

where J_1 and J_2 are the jacobians w.r.t. respectively the first and the second inputs of f.

If the error on the robot position Rr^k_i was correlated with the error on the estimate of another object (feature parametric representation S or P, landmark or area position R) represented by a random vector X_m, then, this correlation must be updated using :

$$C(Rr^k_{i+1}, X_m) = J_1 C(Rr^k_i, X_m).$$

3.2 Introducing new objects

New object can be introduced in the state vector; it will be a new feature at the landmark level, a new landmark at the area level, a new area at the environment level. For example, the area model must be updated when a new landmark has been located w.r.t. the robot frame after the $(i)^{th}$ motion. The position Rl^a of this landmark w.r.t. the current area, can be expressed from its position Rl^r w.r.t. the robot frame, by the use of the robot position:

$$Rl^a_i = f(Rr^a_i, Rl^r) \tag{2}$$

and the error propagation gives:

$$C(Rl^a_i) = J_1 C(Rr^a_i) {J_1}^t + J_2 C(Rl^r) {J_2}^t$$

Since this object is correlated to the robot, it is also correlated with all objects X_m that are already correlated with the robot:

$$C(Rl^a_i, X_m) = J1 C(Rr^a_i, X_m)$$

We have illustrated on figure 2-b (ellipses represent position uncertainties) that uncertainties on landmark positions w.r.t. an area frame, are *always* greater than the uncertainty on the robot position. This implies that the error on the robot position *cannot go below* its initial value; but, if perceived several times, the uncertainty on the landmark position will decrease and the knowledge on the robot position can be improved.

3.3 Adding constraints

A relation detected between the map elements can be expressed by a numerical constraint. Such a relation can arise when the robot measures the position of an already known landmark. For example, after the $(n+1)^{th}$ displacement, the robot localizes the landmark l in the position Rl^r w.r.t. its frame; if it is matched with a landmark already known in the area model with its estimated position Rl^a w.r.t. the area frame , the added constraint can be written :

$$Rl_n^a - f(Rr_{n+1}^a, Rl^r) = 0 \tag{3}$$

The Extended Kalman filter will be employed at this step to minimize this non linear measurement equation. We use the relocation-fusion strategy proposed by Moutarlier and Chatila [6]; the relocation step provides firstly a new estimate of Rr_{n+1}^a, so that we can eventually reduce a bias which could corrupt the error on the robot position. Then, the fusion step computes the new estimate Rl_{n+1}^a of the landmark position w.r.t. the area frame. These corrections will be yet propagated to the other elements of the state vector.

4 Landmarks

The modeling function is based on sensory data acquired by an LRF which executes only a 2π pan scanning on the environment; 2D segments are extracted from the set of acquired points by a classical segmentation algorithm [6]. A description of a line segment S_l is composed by several informations useful during the matching or the fusion process:

P_1, P_2:	end points.
θ, l :	orientation and length.
p,q,type:	minimal parametric representation, with the line equation: $x = py + q$ if $type = 0$. $y = px + q$ if $type = 1$.
$\sigma_p^2, \sigma_q^2, \sigma_{pq}$	uncertainty matrix on $(pq)^t$.

Let us recall that, with a feature-based localization function [6], the estimate of the robot position after the $(n+1)^{th}$ displacement, $\hat{Rr}_{n+1}^g$, can be corrected from at least two matchings between perceived segments known w.r.t. the robot frame S^r and segments which are already known w.r.t. the global frame S_n^g. The measurement equation is:

$$S_n^g - g(Rr_{n+1}^g, S^r) = 0$$

where g is a specific function which transforms the parametric representation of a segment from a reference frame (robot frame) to another (global frame).

In this paper, a landmark-based approach is proposed in order to deal with the relocation problem. A landmark (L) is defined as a set of line segments $L = \{S1, S2, .., Sn\}, n \geq 2$. These segments are connected or not (for example, connected for a corner between walls, not connected for a door). The first and last ones can be endless (only one vertex has been actually perceived). On figure 3, some geometrical landmark shapes are presented; the inclined dashes show where is the material (in the model, the segments are oriented so that this information can be deduced).

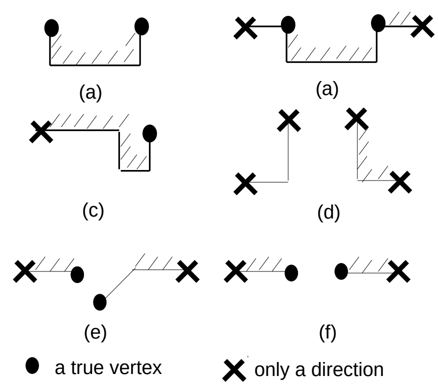

Figure 3: *Different landmark models*

Each landmark has an intrinsic reference frame in which its geometrical model is known. The landmark extraction procedure must tackle a difficult problem: how to anchor this intrinsic frame from the extracted features? On the figures 5 and 4 we present our current choice, when the initial perception of the landmark is a corner; according to the material position w.r.t. the segments (convex or concave) and to the length of the two segments, some simple rules are applied to anchor the landmark frame. Similar simple rules have been defined for a door landmark.

After the definition of the landmark frame, its geometrical description in its proper frame is found. For example, for configuration presented on the left side of the figure 5, the segment supporting the X_l axis is represented by $(p = 0, q = 0, type = 1)$, and the other by $(p = tg(\phi), q = 0, type = 1)$. The first one is perfectly known, while the second one has an uncertainty on the p parameter. The last operation for the landmark extraction, concerns the error propagation from the parametric representations of the perceived segments (4 parameters for 2 segments in our example) w.r.t. the robot frame, to their representations in the landmark frame (1 parameter) and to the landmark position w.r.t. the robot frame (3 parameters).

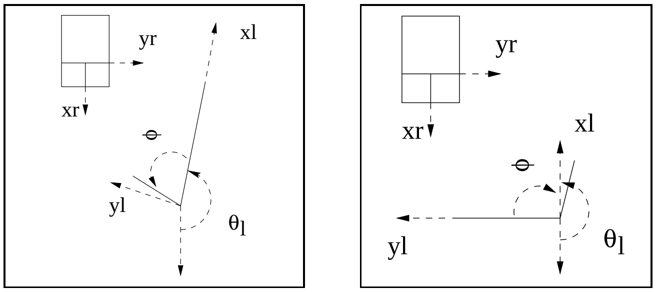

Figure 4: Concave corners.

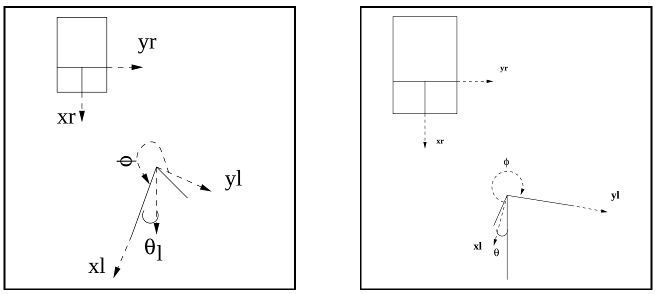

Figure 5: Convex corners.

While navigating, the robot can take advantage of the current knowledge on landmarks in order to select, depending on its position and its corresponding uncertainty, the best landmark (or set of landmarks) it must perceive in order to make easier the localization process [1] [3]. This implies that landmarks must be static, so that they can be perceived at specific positions.

The landmark extraction procedure must provide the landmark position w.r.t. the robot frame: R_l^r and an uncertainty matrix $C(R_l^r)$ which reflects the propagation of the sensory errors on this information.

5 Management of a Hierarchical Model

It is well known that map building and localization processes are not independent (bootstrapping problem [7]). This requires the following two constraints : the robot must have a quite good start position, and the robot must be able to localize itself *regularly* on known landmarks and extract new ones as it explores new unvisited areas.

How can we update our hierarchical model, while the robot is moving in a current area? We summarize hereafter the actions we must undertake on the three model levels (landmarks, areas, environment), when a new event happens: an odometric measurement (update the robot position- see section 3.1), a new object that must be added to a model or a new constraint.

5.1 Update the landmark models

For each landmark, a state vector gives, according to the landmark frame, the robot position as a first element, and then, the parametric representations of each feature (segment or point) which have been associated to this landmark.

After the $(n+1)^{th}$ displacement, the new robot position Rr^l_{n+1} can be estimated by the equation 1 applied to the robot position w.r.t. the landmark frame, initialized when the landmark is discovered by the inversion of Rl^r. Note that we need here the inverse of the transform R_r^l computed by the landmark extraction procedure.

A new feature can be added to a landmark by the use of the following equations:

$$S^l_{new} = g((R^l_r)^{-1}, S^r) \; ; \; P^l_{new} = h((R^l_r)^{-1}, P^r)$$

where S^r and P^r stand for a segment or a point extracted from the sensory data and known in the robot frame, g transforms the segment representation from a reference frame to another one, and h is the transform function for a point feature.

A constraint must be satisfied when a feature of a landmark is perceived again; Kalman filter is applied on the following equations, where S^l_n and P^l_n are a segment and a point already integrated in the landmark model, and S^r and P^r are the corresponding matched features of the $(n+1)^{th}$ perception:

$$S^r = g(Rr^l_{n+1}, S^l_n) \; ; \; P^r = h((Rr^l_{n+1}, P^l_n)$$

Rr^l_{n+1} represents the estimated robot position from the odometric measurement, before any relocation. The relocation-fusion method is applied in order to correct at first Rr^l_{n+1} before any correction on the geometrical description of the perceived landmarks.

5.2 Update the area models

For each area, the state vector gives, according to the area frame, the robot position as a first element, and then, the positions of the landmarks discovered in this area.

The robot position can be updated by the equation 1 applied to the robot position w.r.t. the area frame which is initialized when the robot enters the area through a door landmark.

A new landmark can be added to the area model, by the use of the equation 2, where Rl^r is directly given by the landmark extraction procedure. In the same way, the equation 3 is used to update the robot position w.r.t. the area frame and the landmark positions when an already perceived landmark is localized again.

5.3 Update the environment model

For the environment, the state vector gives, according to the global frame, the robot position as a first element, and then, the positions of the areas successively crossed by the robot.

The robot position can be updated by the equation 1 applied to the robot position w.r.t. the global frame; this position is initialized by an high level decisional system, according to the task the robot must perform.

A new area can be added to this environment model; this event happens when the robot detects a door landmark and crosses the door. In our current system, the door reference frame is used as the reference frame for the new area (see figure 6): this solution can be modified in the future because experimental results have shown that a door positioning is less accurate than a corner positionning for example. The following equation is used in order to initialize the spatial relationships between a new area and the global frame:

$$R^g_{newarea} = f(R^g_{currentarea}, f(Rr^{currentarea}, R^r_{door}))$$

If the robot enters an area always by the same door, a new constraint can be added on the same way than for the area model (see equation 3). But, at this time, we have not validated by simulation or actual experiment, the relocation-fusion procedure that must be executed on the environment model when the robot finds that it has returned in an area by another door; on the figure 1, this event happens if the robot discovers the area $\mathcal{A}_1$ through the door $P0 \leftrightarrow 1$, then explores $\mathcal{A}_2$ and $\mathcal{A}_3$ and at last, returns to $\mathcal{A}_1$ through the door $P3 \leftrightarrow 1$. Such a scenario is being analyzed by now.

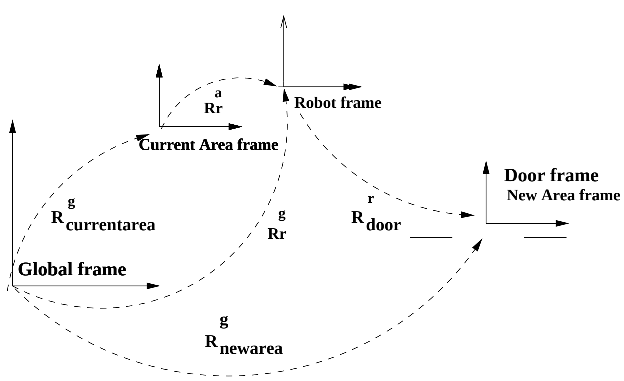

Figure 6: *Management of the topological model*

6 Experimental results

The landmark-based localization procedure has been validated in a number of experiments on our mobile robot Hilare2; this procedure has been integrated on board as a functional module, under the real time environment VxWorks. On the figure 7, the robot has discovered two areas: our experimental room and a corridor. A reference map of our environment has been superposed to the global model built after 7 acquisitions so that the consistency of the perceptual model is visualized.

The figure 8 gives the area model of the experimental room after 11 perceptions. Ten landmarks have been extracted from the sensory data, and the robot position has been corrected after 9 elementary displacements from one, two or three matchings between already discovered landmarks and perceived ones; on the figure 9, the landmarks extracted by the relocation procedure are drawn, with their view field centered on the sensor frame.

The matching procedure has failed only after displacement from position 5 to position 6. Other landmarks are discovered from the positions 6 to 9 in the left side of the room, so that the robot position can be updated, but the consistency with the landmarks found before position 6, has been lost. So, when the robot returns in the right side of the room after the last displacement, the robot position w.r.t. the area, had a bias with almost 40cm in translation and 5 deg in rotation. Nevertheless, the last relocation is successful and the robot position can be corrected so that the final model is consistent.

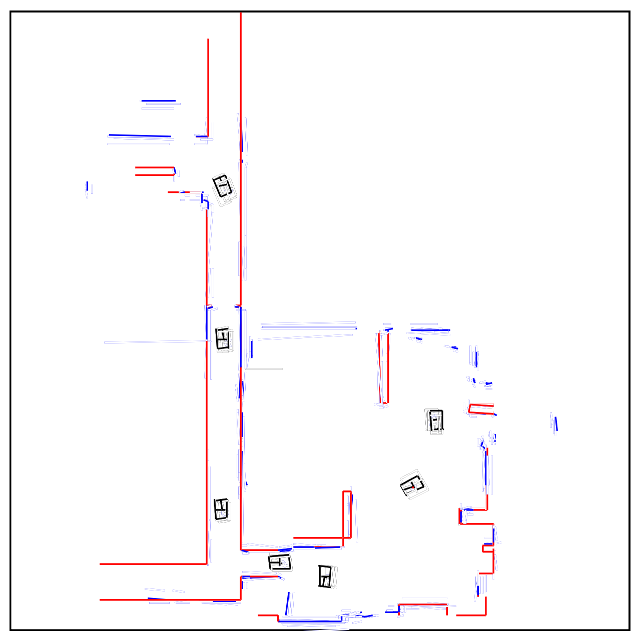

Figure 7: *An experiment with two places*

7 Conclusion

In this paper, we have addressed the self-localization and world modeling problems for a mobile robot which

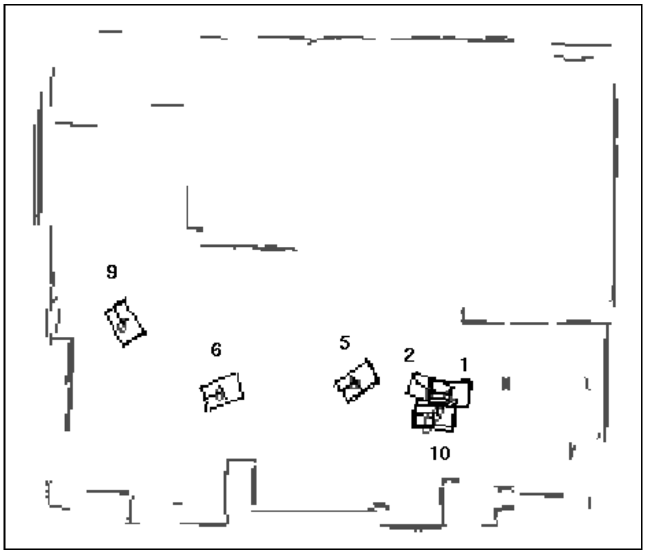

Figure 8: *Final area model after 10 displacements*

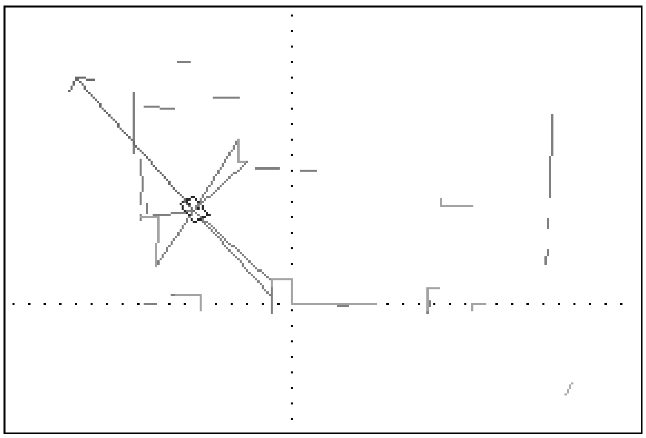

Figure 9: *Landmark matchings at step 9*

must explore an indoor environment, without any a priori knowledge. We cope with the relocation-fusion problem: the classical feature-based approaches lead to the construction of very cumbersome geometrical models and to a combinatory explosion if some structuring or forgetting rules are not applied.

We have proposed such strategies, so that the complexity of the world model is controlled while the robot discovers the environment. First of all, only useful perceptual groupings of features, referred as landmarks, are extracted from each sensory data; a discriminant utility function must be applied, so that unuseful features are forgotten. The localization of landmarks w.r.t. the robot frame, provides spatial relationships between the robot and the environment, from which we can update the robot position and the landmark descriptions (position and structure).

Then, another issue is the decomposition of the world model in several levels, based on topological rules: correlations between landmarks are only taken into account between the ones which can be perceived from the same robot position, i.e. for indoor scenes, between landmarks extracted in the same area (room, corridor,...) Experiments are on the way in order to validate the topological modeling and to provide a quantitative evaluation of the model accuracy.

Acknowledgements The authors wish to thank S.Betgé-Brezetz and P.Hébert for numerous discussions about object-based and topological modeling.

References

[1] R. Bauer. Active Manoeuvres for Supporting the Localisation Process of an Autonomous Mobile Robot. In *Proceedings of the Second International Workshop on Intelligent Robotic Systems (IRS'94), Grenoble (France)*, Juillet 1994.

[2] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation, St Louis (USA)*, Avril 1985.

[3] M. Devy and H. Bulata. Landmark-based vs Feature-based Localization of a Mobile Robot in a Structured Environment. In *International Conference on Advanced Robotics (ICAR), Barcelone (Spain)*, 1995.

[4] P. Fillatreau and M. Devy. Localization of an autonomous mobile robot from 3d depth images using heterogeneous features. In *IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama, , Japan)*, Juillet 1993.

[5] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic Map Building for an Autonomous Mobile Robot. In *International Journal of Robotics Research*. Massachusetts Institute of Technology, August 1992. Vol. 11, No. 4.

[6] P. Moutarlier and R. Chatila. Incremental free-space modelling from uncertain data by an autonomous mobile robot. *IEEE International Workshop on Intelligent Robots and Systems (IROS'91), Osaka (Japon)*, Novembre 1991.

[7] W.D. Rencken. Concurrent Localisation and Map Building for Mobile Robots Using Ultrasonic Sensors. In *IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama, (Japan)*, 1993.

[8] R.C. Smith and P. Cheeseman. On representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4), Winter 1987.