Proc. of the 3rd International ICSC Symposium on Soft Computing (SOCO'99)
June 1999, Genova, Italy, pages 754-760

Integrating fuzzy geometric maps and topological maps for robot navigation

Jorge Gasós
IRIDIA, Université Libre de Bruxelles
50 av. F. Roosevelt, CP 194/6
B-1050 Brussels, Belgium
jgasos@iridia.ulb.ac.be

Alessandro Saffiotti
Applied Autonomous Sensor Systems
Dept. of Technology, University of Örebro
S-70182 Örebro, Sweden
alessandro.saffiotti@ton.oru.se

Abstract

Autonomous mobile robots need to use spatial information about the environment in order to effectively plan and execute navigation tasks. This information can be represented at different level of abstractions, ranging from detailed geometric maps to coarse topological maps. Each level is adequate for some sub-tasks, but not for others. In this paper, we propose to use *hybrid maps*, patchworks of local metric maps connected into a topological network. Our hybrid maps are peculiar in that they use fuzzy sets to represent the uncertainty that affects metric information. We show how a robot can build hybrid maps from sensor data, and how it can use them in autonomous indoor navigation. We also report experiments performed on a real mobile robot that demonstrate the robustness of our approach with respect to inaccuracies in the map and noise in the sensor data.

1 Introduction

Robot navigation in large scale indoor environments requires an adequate representation of the working space. This representation should be abstract enough to facilitate higher level reasoning tasks like strategic planning or situation assessment, and still be detailed enough to allow the robot to perform lower level tasks like trajectory generation or self-localization.

A common belief in the robotics field is that robots need to represent and reason about information at different levels of abstraction at the same time [4; 5]. There are several reasons for this. First is epistemic adequacy: different tasks call for different types of representation. For example, global navigation strategies are more easily planned using a topological map, where we can decide the sequence of rooms and corridors to be traversed; but fine motion control needs geometric information to precisely control navigation among features and obstacles. Second, computational adequacy: geometric information is difficult to collect and expensive to handle, and we cannot pay the price to maintain a detailed geometric representation of the entire environment where the robot can operate. The final reason is ontological adequacy: fine grained information is difficult to obtain *a priori* and is likely to change with time; coarse maps are easier to estimate and more prone to remain valid over time.

In this paper, we consider the representation of spatial knowledge at two different levels of abstraction which are commonly considered in the robotics literature: the geometric level, and the topological level. We propose to represent the environment by *hybrid maps*, patchworks of local metric maps, called *sectors*, connected into a topological network. A distinctive feature of hybrid maps is that they do not need to be metrically consistent on the global scale — although they are metrically consistent locally, in each sector. The hybrid structure allows us to combine abstract global reasoning and precise local geometrical computations. Moreover, this structure reflects the typical organization of indoor environments, where rooms and hallways define independent but connected local working spaces. To navigate in the environment, the robot uses the topological information to plan a sequence of sectors to traverse, and uses the metric information in each sector to locally move within the sector and to the next one.

In the rest of this paper, we describe our technique for representing hybrid maps. We put a particular emphasis on the problem of how to accommodate the uncertainty and imprecision that inevitably affects the information available to the robot. Our approach to represent uncertain spatial information is based on the theory of fuzzy sets, and builds on our previous works [2; 9]. We also show the uses of hybrid maps for the generation and execution of navigation plans, and illustrate these uses with experiments of indoor navigation performed on a real mobile robot. In these experiments, the topology of the hybrid map is given *a priori*, while the geometrical information in each sector is automatically acquired by the robot. The experiments stress the robustness of our approach with

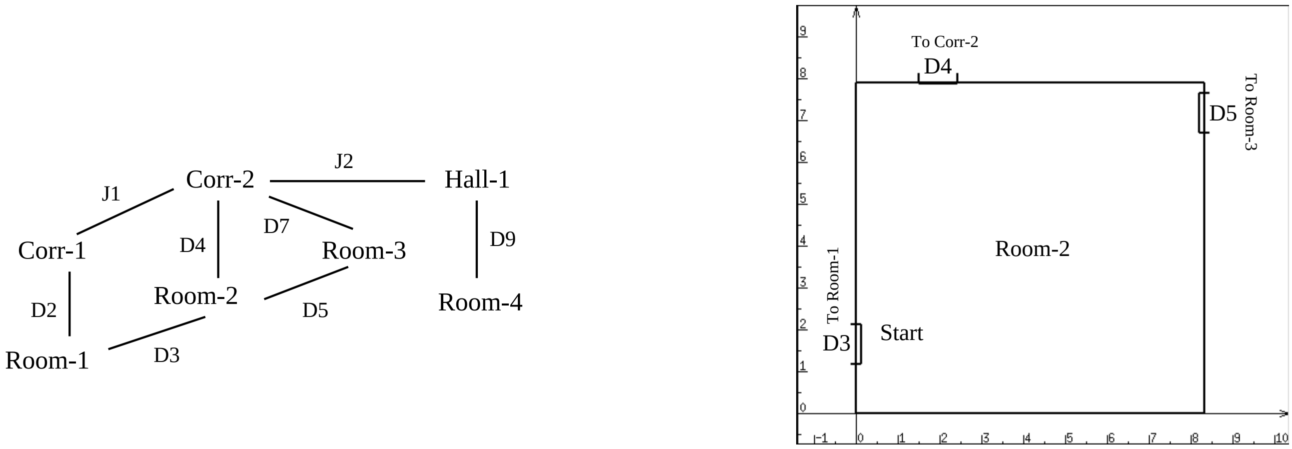

Figure 1: The topological map of IRIDIA (left), and the metric map of one sector (right).

respect to inaccuracies in the map and errors in the local position estimation.

2 Hybrid maps

Hybrid maps integrate environment information at two distinct levels of abstraction: (i) local geometric information, encoded in a set of local sectors; and (ii) global topological information, encoded in a network connecting these sectors.

Each sector is a Cartesian representation, with its own reference system, that covers a limited area of the environment, like a room, a hall, or a corridor. Each sector includes an approximate geometric description of the boundaries of (the objects in) the environment, represented by a set of *fuzzy segments*. This geometric description is initially empty since we assume no previous knowledge at the level of objects in the sectors. As the robot navigates in a given sector, consecutive measurements of the ultrasonic sensors provide information about the objects located in the proximity of the robot trajectory. Analyzing and aggregating these measurements we build a polygonal approximation that corresponds to the 2-D ground projection of the sector outline. This is a compact representation that approximates the boundaries of the objects, defines their spatial location, and allows fine motion control.

Beside the set of fuzzy segments, each sector also contains a set of *landmarks*, which specify the position and orientation of some objects that are relevant to robot navigation. Typical landmarks include walls, doors, and corridor junctions. Since these are intrinsic parts of the environment that do not change with time, we assume that approximate prior information of landmark location is provided to the system. Some of these landmarks, like doors and junctions, can be shared by two sectors, just acting as *gateways* from one sector to the other. The gateways encode the information about the connectivity between sectors; a *hybrid map*, then, is a topological network where nodes are sectors and arcs are gateways.

Fig. 1 shows (a part of) the hybrid map for the IRIDIA lab that we have used in our experiments. The left hand side shows the topological network of sectors (nodes) and gateways (arcs). The right hand side shows one of the sectors (Room-2). The sector contains its own coordinate system, and the available metric infomation, which in this case is limited to the prior knowledge about the position of walls and doors (gateways), plus the robot's starting position. We shall show below how the robot can complete this map by building the fuzzy segments.

In a sense, a hybrid map contains global metric information. In fact, since we know the position of each gateway in each one of the sectors that it connects, we can easily compute the coordinate transformation from any sector to each one of its neighbors. By iterating this computation, all the metric information of the entire map could be expressed with respect to a global reference system. However, the metric information about the position of gateways (and of landmarks in general) is approximate, a large amount of error can be produced in the propagation of geometrical information from one sector to the next. One consequence of this is that if there is more than one path connecting two sectors in the topological graph, then we may get different coordinate transformations between these sectors by choosing different paths. The key point of hybrid maps is that the global map does not need to be (and in general it is not) metrically consistent, although each local map is.

3 Using hybrid maps

Hybrid maps have several uses, including path planning, map building, and exploration of partially unknown or modified environments. In this paper, we

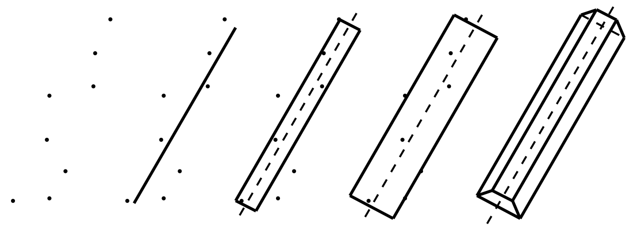

Figure 2: Building a fuzzy segment from a set of scattered points.

focus on the use of hybrid maps for the generation and execution of navigation plans. At the higher level of abstraction, we use the topological network in the hybrid map to plan a sequence of sectors and gateways to traverse, and to guide execution by activating a corresponding sequence of navigation behaviors. At the lower level, we build the local maps and use them to perform geometrical navigation within each sector, and to perform sensor-based self-localization.

3.1 Building the sector maps

Ultrasonic sensor observations, acquired as the robot moves, can be used to build geometric maps of indoor environments. Based on the estimation of the robot position provided by the dead reckoning system, sensor measurements can be transformed into a discrete set of points that correspond to object locations in the sector coordinate system. Analyzing and aggregating these measurements we can recover the outline of the objects detected by the sensors: a sequence of smoothly changing observations indicates that the side of an object is being detected; a discontinuity that starts a new smooth sequence indicates that a new object has been detected; and a single measurement, not related to its neighbors, can be considered noisy and discarded. The approach is based on the following sequence of steps [2]:

- sensor measurements are preprocessed to eliminate easily detectable misreadings and to group together smoothly changing consecutive measurements from the same sensor;
- each group of measurements is fitted to a straight line (segment) and the uncertainty on its position is represented using fuzzy sets; and
- fuzzy segments corresponding to the same side of an object are merged to obtain a single boundary. Uncertainty is propagated from the fuzzy segments to the fuzzy boundaries.

Uncertainty representation is a key aspect of this approach. Boundary extraction from sensor observations is a process that works with inaccurate and noisy data, hence, generating uncertainty on the position, orientation and size of the boundaries. Given a finite set of obsevations coming from the same side (boundary) of an object, we approximate them by a straight line while representing the uncertainty in the knowledge of its real position by a fuzzy set (Fig. 2). The degree of membership of a point to the fuzzy set indicates the possibility that this point belongs to the object detected by the sensor. Extending this definition to the case of lines (boundaries), the degrees of membership can also be interpreted as degrees of similarity when two different boundaries are compared. Thus, a low (high) overlap of their membership functions will indicate that they should be considered similar to a low (high) degree. This representation allows to express the information/uncertainty contained in the sensor data and is closer to the main operation performed on the boundaries: deciding if boundaries obtained from different sensors or from different positions correspond to the same object.

The result of applying this map building process in a sector is a collection of fuzzy segments that provide approximate information about the position, orientation and length of the boundaries of the objects.

3.2 Topological planning on hybrid maps

At a higher level of abstraction, the robot needs to develop navigation plans to move from one point in the environment to another. In large, dynamic and partially unknown environments, computing a detailed collision-free path in the metric space is not feasible, as the position of all the static and moving obstacles in the environment cannot be anticipated at planning time. A more common approach is based on the concept of behavioral navigation: the robot only plans a sequence of abstract navigation sub-tasks that need to be executed, but does not specify the details of execution. For example, a plan to get from the "start" position in Fig. 1 to an information desk in Hall-1 may be to first *get-close-to* door D4, then *cross* it, then *follow* Corridor-2 up to junction J2, then *traverse* J2, and finally *get-close-to* the desk. Each sub-task is performed by a specialized "behavior," with concurrent behaviors taking care of real time obstacle avoidance. The actual trajectory followed by the robot depends on how the activated behaviors respond to the environmental contingencies encoutered during execution.

A sequence of navigation behaviors can be easily planned using the topological information in a hybrid map. The task of a topological planner is to find the best (according to some cost criterium) path in the topological graph going from the sector that contains the current location of the robot to the one that contains the goal. Then, the planner decides which behaviors must be used to cross each gateway in the path, and to navigate from one gateway to the next one inside each sector. To do this, the planner needs to know which behaviors in the robot's repertoire can be used to achieve a given sub-goal. For example, the "cross" behavior can be used when in front of an open doorway to achieve the goal of being on the sector on the other side; and the "follow" behavior can be used to go down a corridor until a desired gateway is reached. The details of our technique for producing behavior-based navigation plans are given elsewhere [8]; here, we only emphasize that using this type of planning requires that abstract, topological information about the environment is available.

3.3 Navigating on hybrid maps

In order to execute the navigation plans, the robot needs to keep track of its own position in the map as it moves. In a hybrid map, this position is given by two components: (i) a pointer to a sector in the topological network; and (ii) a position estimation in the Cartesian coordinate system of that sector. The position within the sector is continuously updated based on odometry and on the observation of perceptual features. Updating the global pointer within the network relies on gateway traversal.

More precisely, self-localization inside a sector is based on the matching of maps that were built by the robot in different moments (see [2] for details). Each sector is accociated to a *sector map* made of fuzzy segments, and to a coarse ($1\,m \times 1\,m$ cells) binary grid that records information about which areas in that sector have been already explored by the robot. When the robot enters a sector for the first time, both the sector map and the sector grid are empty. While the robot moves through not visited cells, it adds the new segments to the sector map, and marks the corresponding cells in the grid as visited. When it arrives to a visited area, the newly built segments are added to a *local map*. The local and sector maps are then compared until a significant number of local segments match those in the sector map. Then, the transformation that brings both maps together is used to estimate the dead reckoning errors accumulated in the time span between the construction of the two maps. Once these errors are corrected, the uncertainty about the robot's location is bounded by the uncertainty of the sector map, that depends on the size of the environment and on the precision of the dead reckoning system. In this way, by continuously re-localizing, the robot can navigate for long periods of time without the help of external positioning systems. This approach also allows to map the environment in an incremental way as new data appear in the local maps; it also allows the robot to adapt the map to changes in the layout that may occur in moderately dynamic environments.

Localization in the topological network is based on gateway traversal. It requires detecting that the robot is crossing a gateway and updating its position

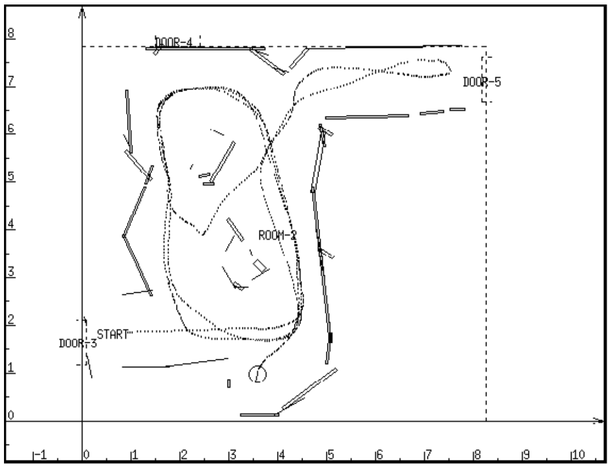

Figure 3: Fuzzy segments built by the robot in the map building experiment.

in the network. Gateway detection is based on the knowledge of its approximate position in the sector and confirmed by the sensor observations. Each type of gateway (i.e., door, corridor junction) has a particular shape and will generate a different distribution of observations (signature) when detected by the robot sensors. Once a gateway has been crossed, the topological network indicates the new sector and the position of the robot in its local coordinate system.

4 Experiments

In this section we provide two illustrative examples of the results obtained by the system in the navigation of a mobile robot in indoor office environments. The first experiment presents the process of map building in a previously unknown environment. The second focuses on path planning and navigation using hybrid maps. The experiments were conducted on a Pioneer mobile robot equipped with seven Polaroid ultrasonic sensors and with wheel encoders for position estimation. The robot uses a fuzzy behavior-based navigation system (including point to point movement, door crossing, obstacle avoidance, and contour following) for navigation [8].

The experimental site was the IRIDIA laboratory, that is composed of a number of rooms and corridors connected by doors and junctions (a part of it is shown in Fig. 1). A priori information included the topology of the environment (sectors and gateways connecting them) and, for each sector, the approximate position of intrinsic parts of it (walls and doors).

4.1 Map building

Geometric information about objects in the sectors is automatically acquired by the robot using the map

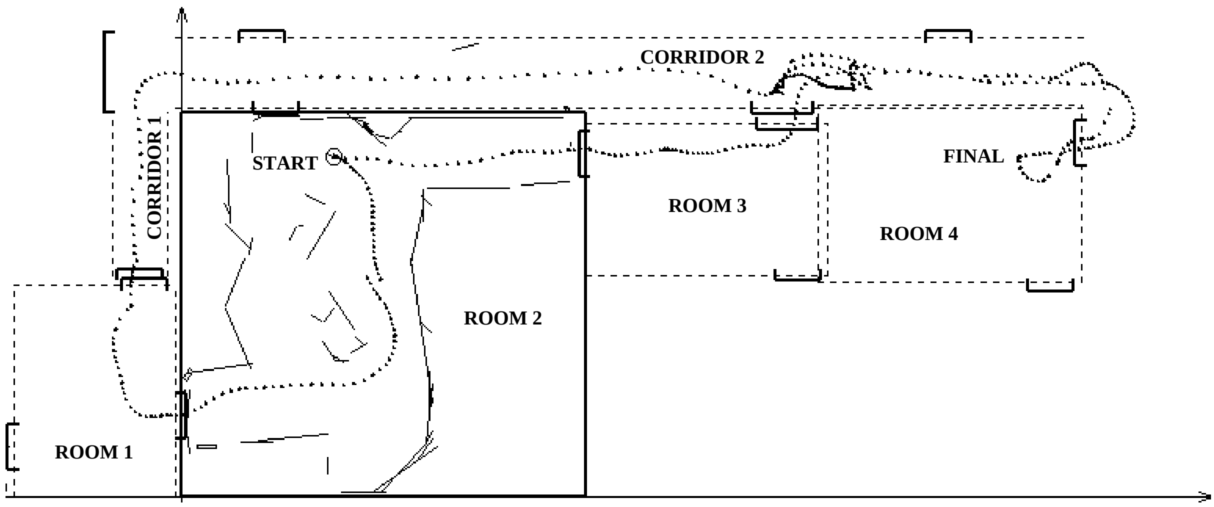

Figure 4: Robot's trajectory using self-localization

building facility presented in Sec. 3.1. In our first experiment, we have let the robot explore the sector shown in Fig. 1 (right). The environment is about 8×8 meters. At the beginning of the experiment, the details of the sector layout are unknown to the robot. The robot moves through the room using a simple exploration behavior (random wandering) and collecting sensor measurements. The system detects sequences of aligned observations that, once the uncertainty on their real location is considered, are used to build the fuzzy segments. Fig. 3 shows the α-cut in 0 of these fuzzy segments at the end of the exploration, together with the robot's trajectory. The robot travelled about 50 meters at an average speed of 110 mm/sec; the entire exploration took slightly less than 8 minutes.

As it appears, the fuzzy segments provide a good polinomial approximation of the spatial layout of the environment. The outline of the objects with compact and flat surfaces was obtained, and the discontinuities in their shape have been detected. The gaps in the model correspond to areas that were occluded by other objects, to sides of objects with an orientation perpendicular to the robot trajectory, and to objects with very irregular surfaces.

It should be noted that the robot uses the self-localization process described above when it returns to an area that has been previously explored. If it didn't, due the cumulative error of odometry the newly perceived segments would not match the previously built one in the same area, and the map would quickly become cluttered with distinct segments corresponding to the same objects. By using self-localization, the error in the map is bounded by the size of the sector — i.e., by the space that the robot must travel relying on odometry alone before it can re-localize. The need to keep this error small justifies the use of small, local sectors.

4.2 Path planning and navigation

Our second experiment involves planning and navigation in the topological network. The task for the robot was to go from the "start" position to the "final" position (see Fig. 4) and back. The best path found by the planner was to go from Room-2 to Room-3, then to Corr-2 and then to Room-4, where the goal was located. (The planner did not chose to go through D4 as it was know to be closed.) To return to the starting position the robot tried the same path, but in the meanwhile the door between Room-3 and Corr-2 (D7) had been closed. After some manouvers in front of this door, the robot detects that it is closed and plans an alternative path to return.

Fig. 4 illustrates the result of this experiment, as drawn on the graphical interface of our robot. The figure shows part of the hybrid map used by the robot in this experiment. The *current sector*, the one where the robot currently is, is indicated by thick borders; the shown coordinate system is the one local to the current sector. Sectors that are part of the navigation plan but are not current are indicated by dashed lines; for graphical clarity, only the fuzzy segments in the current sector are drawn. Gateways between sectors (here, just doors) are marked by thick lines. Note that some doors are drawn twice at two slightly different positions. As noted in Sec. 2, this is an effect of the inaccuracies in the metric information regarding the position of the gateways — recall that a hybrid map does not need to be globally consistent.

The figure also shows the trajectory travelled by the robot according to the estimates provided by the auto-localization process. The overal trip was about

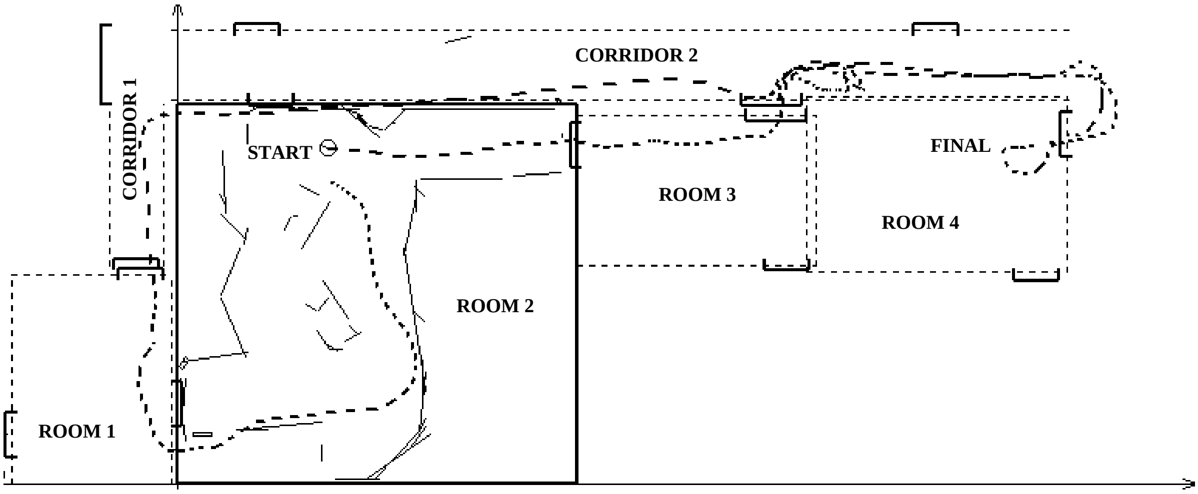

Figure 5: Robot trajectory using internal odometry

50 meter long and it took about 7 minutes. Because self-localization is performed continuously, its effect is usually not apparent in the trajectory. When re-localizing after entering a new sector, however, initial self localization may produce a larger update. These updates produce visual discontinuities in the trajectory, for instance in the middle of Corr-1, and of Room-2. The messy trajectory near the closed door D7 corresponds to the tries the robot did to cross this door until it decided that it was not able to traverse it and planned an alternative path.

The positive effect of the self-localization process is best appreciated if we compare the previous figure with Fig. 5. This picture shows the trajectory travelled by the robot as measured by its own internal odometry, which was recoded during this run. (The robot's behaviors, however, were controlled by the corrected estimates shown in Fig. 4.) Due to the large amount of turning, especially when checking the closed door, the odometric data strongly diverge from the corrected estimates. Those data also strongly diverge from the real position of the robot: to see this, note that according to these data the robot has passed through a couple of walls (see Fig. 5), which it obviously didn't. In quantitative terms, at the end of the run the position estimated by our self-localization process was 3.5 cm off from the real robot's position, measured by hand, while the position measured by the robot's internal odometry was 45.8 cm off from the real one.

4.3 Computational effort

Due to the hierarchical organization, hybrid maps have nice computational properties. Navigation on the topological graph is inexpensive since the graph is coarse-grained. Fine-grained metric localization is computationally expensive in general, but becomes tractable in our case because we focus the attention to the current sector. Moreover, the use of fuzzy sets to represent the positional uncertainty of segments simplifies the computation by reducing the correspondence problem to the search for fuzzily overlapping segments. Only when the localization error is large, resulting in few or no overlaps, we need to engage in a full search for a best correspondence. Since we relocalize the robot very often, this is rarely needed.

In our implementation, we have included the self-localization routines in the main 100 msec control loop of the robot. Not all the routines are called at every cycle: for example, the segment building process is only called every 10 cycles in order to leave the sonars enough time to collect some new data; and the matching between the local and sector maps is only attempted whenever some new segments are produced (every 25 cycles on the average). The following table reports the CPU times measured on a Sparc 20 machine, in milliseconds.

Process	*avg time*	*freq*	*time/cycle*
Collect data	0.8	10	0.8
Build segments	17.2	1	1.7
Match maps	145[1]	0.4	5.8

The columns show: the average CPU time used by the process each time it is called; the frequency at which the process is called, in Hz; and the average time per control cycle used by the process.

[1]Since map matching may require more than the 100 msec available within one cycle when it must search for a full corespondence, we have divided its computation over several successive control cycles.

5 Conclusions

The use of hybrid representations for spatial knowledge is gaining popularity in the field of mobile robotics [4; 5; 1]. These representations can accommodate and integrate different aspects of knowledge at different levels of abstracion. The work presented here belongs to this trend. Our hybrid maps, however, are peculiar in that we use techniques based on fuzzy sets to represent the uncertainty that affects metric information. Although others have used fuzzy sets to represent approximate geometric maps [11; 6; 7], we are not aware of any other approache in which these maps are integrated with abstract topological maps. We have reported experiments that demonstrate the robustness of our maps with respect to inaccuracies in the map and noise in the sensor data. In an extended report we show that our approach also tolerates major errors in the odometry, like those produced by a large slip in one wheel [3].

Our technique is related to the work presented by Thrun and collegues in [10], one of the main differences being that we use the theory of fuzzy sets, while that work is based on probability theory. The use of fuzzy sets gives us two main advantages. First, thanks to the qualitative flavor of fuzzy sets, we could develop our technique using heuristic knowledge and without the need for complex probabilistic models. Second, the use of fuzzy segments simplifies the matching problem, resulting in good computational efficiency. As we have shown, we can perform incremental map building and continuous self-localization in real time with a computational overhead of the order of milliseconds, independently on the size of the environment. This contrasts which Thrun and collegues' technique, that reportedly required almost one hour of computation on two machines to generate the largest maps.

The major current limitation of our approach is that we only take uncertainty into account at the geometric level, not at the topological level. At every given moment, the robot has a crisp idea of which sector it is in. If this estimate is wrong, the robot is hopelessly lost, since metric self-localization will only be attempted inside the current sector. Modeling the uncertainty at the topological level is necessary to further increase the robustness of our approach, and it is a major priority of our current work.

Acknowledgements The first author was supported by a Marie Curie grant, founded by the EC. Support to the second author came partially from the Swedish KK Foundation.

References

[1] S.P. Engelson and D. McDermott. Error corection in mobile robot map learning. *Proc. of the IEEE Int. Conf on Robotics and Automation.* Nice, France, 1992, pages 2555–2560.

[2] J. Gasós and A. Rosetti. Uncertainty representation for mobile robots: perception, modeling and navigation in unknown environments. *Fuzzy Sets and Systems.* Forthcoming. Available on-line at ftp://iridia.ulb.ac.be/pub/jgasos/fss99.ps.gz.

[3] J. Gasós and A. Saffiotti. Integrating fuzzy geometric maps and topological maps for robot navigation. *Technical Report.* IRIDIA, Université Libre de Bruxelles, 1998.

[4] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.

[5] B. Kuipers and T. Levitt. Navigation and mapping in large scale space. *AI Magazine*, 9:25–43, 1988.

[6] G. Oriolo, G. Ulivi, and M. Vendittelli. Real-time map building and navigation for autonomous robots in unknown environments. *IEEE T. on Systems, Man, and Cybernetics*, 28(3):316–333, 1998.

[7] A. Saffiotti. The uses of fuzzy logic for autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997. On-line at http://iridia.ulb.ac.be/FLAR/.

[8] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481-526, 1995.

[9] A. Saffiotti and L. P. Wesley. Perception-based self-localization using fuzzy locations. In M. van Lambalgen L. Dorst and F. Voorbraak, eds, *Reasoning with Uncertainty in Robotics*, number 1093 in LNAI, pages 368–385. Springer-Verlag, Berlin, DE, 1996.

[10] S. Thrun, J-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topolotical and metric maps for mobile robot navigation: a statisfical approach. *Proc. of the 15th AAAI Conf.*, Madison, Wisconsin, 1998, pages 989–996.

[11] A. Tirumalai, B. Schunck, and R. Jain. Evidential reasoning for building environment maps. *IEEE Trans. on Systems, Man, and Cybernetics*, 25(1):10–20, 1995.