Robot Navigation in Centimeter Range Labyrinths

G. Caprari, K.O. Arras and R. Siegwart

Institute of Robotics Systems
Swiss Federal Institute of Technology Lausanne (EPFL)
CH–1015 Lausanne

E-mail: gilles.caprari@epfl.ch
http://dmtwww.epfl.ch/isr/asl

Abstract. This paper presents a topological-metric approach to navigation for mobile mini robots (MMRs). Motivated by future applications of MMRs like remote inspection tasks in small pipe systems, we investigate narrow, labyrinth-like environments (corridors width of 3 cm). Experiments in navigation tasks like local localization, global localization and map-building are carried out with our autonomous robot Alice (dimensions 2x2x2 cm). The paper describes the robot, its locomotion, sensors, communication and user interface. We further discuss sensor modeling for odometry and mapping, place recognition and finally their typical limitations for MMRs. The experimental results suggest that even with a robot of limited size like Alice, it is possible to successfully navigate in environments never reachable before. This opens up new applications for mobile mini robots and motivates further research.

1 Introduction

In the last decades many efforts have been done to reduce the size of mobile robots and many research labs around the world have shown smart and impressive results letting us imagine even better ones in the future [1-5]. These developments are driven by a couple of motivations like the excitement for small-size technology, academic research, or international student competitions [6]. Practical motivations are: space limitations for experiments [7], cost issues or even because small robots can not cause damage in case of failure. Industrial interests are currently to find mainly in the toy market because beside other explanations the field of Mini Mobile Robots (MMRs) is not yet mature and not enough well managed. Thus a lot of work remains to be done in order to fulfill “real world” application where MMRs present great advantages.

Good candidates are exploration tasks, inspection of small plants or mapping of environments unreachable by humans. In all three cases, essential work is to gather information with the on-board sensors on the robot and then send these data to the user in a useful form. An important point about the measurement made with the onboard sensor (distance, luminosity, temperature, etc.) is to know the position where it has been done and not only the value itself. The position is essential for further treatments but not at all trivial to obtain, especially in the case of such small robots like Alice. Mobile robots in this size suffer in a particular way from non-systematic odometry errors like slippage and collisions. Methods for local localization, global localization and map building are therefore needed for a mobile robot which is supposed to navigate successfully.

There are different ways to solve the localization problem depending on the constraints of the robot and the environment. When acceptable, an external camera [8] or a GPS-like system [9], can entirely solve the problem. In the case of multi robot missions, measuring the relative distance between teammates and doing triangulation can be an elegant way [10].

MMRs are good candidates for remote inspection tasks, especially in man-made infrastructures where small structural dimensions stand in contrast to a large overall size of the entire system. Examples include building infrastructure like ventilation systems or small diameter pipeline systems. Hardware requirements for such missions are mobility, high autonomy with respect to energy, embarked sensors and communication, and high mechanical and electrical robustness and reliability. On the software side the robot must exhibit partial autonomy for reactive local navigation maneuvers, should be able to navigate globally and ask for help in situations which are beyond its capacities. Having the above applications in mind, we consider in this work structured, labyrinth-like environments as test-bed for Alice's navigation algorithms. They are mostly structured in a way which allows the robot to rely on some environment regularities like corridor width, the existence of distinctive features or angles between two intersecting tracks.
In the next section the entire hardware systems presented. In section 3 the underlying navigation methods will be exposed introducing the work done in relation to localization and mapping. In chapters 4 and 5 the respective results will be discussed before coming to the conclusion.

2 System Description

The system consists of mainly 3 parts: a wheeled mobile robot, a bidirectional radio connection and an host PC running Matlab. The robot operates more or less as a slave, only obstacle avoidance is performed locally with the onboard microcontroller which drives the motors according to the proximity sensors measurements. The necessary data for localization (motor increments and sensors values) are sent to a PC via a radio link. This permits to overcome the limited processing power of the robot’s microcontroller, to easily develop the algorithms on the PC side and to provide a user friendly interface and control panel.

2.1 The Robot Alice

The main advantages of the robot Alice are the small size and the very long power autonomy of 5 hours. Figure 1 shows the robot in its environment and the modular architecture, whereas Table 1 gives a short description of the relevant characteristics. In [11] and in Table 2 you can find further details about the hardware and the basic concepts.
One big limitation with small robots like Alice is the maximal current that the batteries available

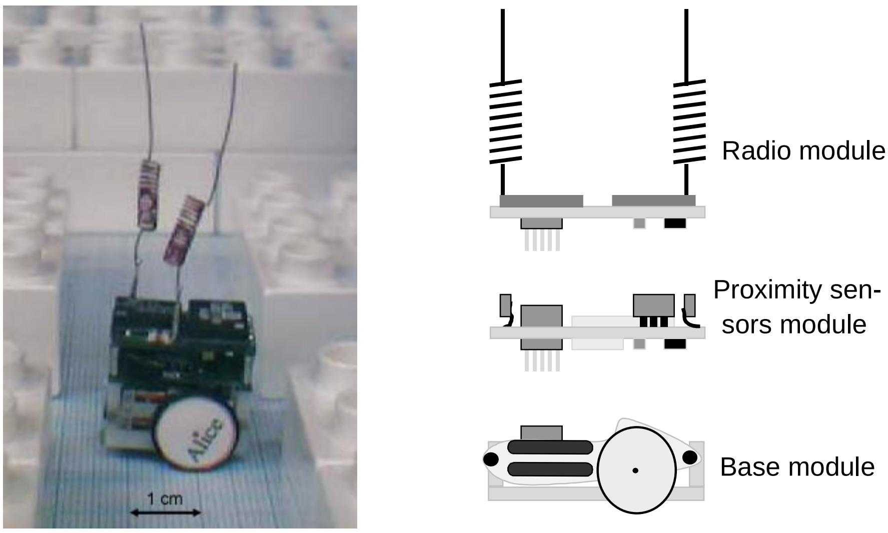

Figure 1. Alice in a 3 cm narrow labyrinth and the composing modules.

in this size (typically silver-oxide button cells or similar) can deliver to the electronics, to the actuators and to the communication devices of the MMR. This imposes severe restrictions to the choice of components used in such small robot and asks for smart or even drastic solutions to reduce the mean and peak current consumptions. Among these: slow communication data rate and CPU clock; slower motor speed or stepwise movements; slow refresh of sensor measures; sequential against parallel transmission, measure and locomotion; and simplified algorithms.

Dimension	21 x 21 x 22 mm^3
Weight	8 g
Velocity	40 mm/s
Power consumption	9 - 18 mW
System autonomy	up to 5 hours
Proximity sensor range	30 mm
Radio communication	10 m, 1000 bps

Table 1. Characteristics of the robot Alice in the experiments configuration.

Microcontroller	PIC16F84 (8 bit CPU, 68 RAM, 1KWord ROM)
Power supply	3 button cells V377
Actuators	2 bidirectional Swatch motors
Sensors	4 IR proximity sensors: SFH900
Radio receiver	RX1020 @ 433.92 MHz by RFM
Radio transmitter	HX1000 @ 433.92 MHz by RFM (on-off keyed)

Table 2. Principal hardware components of the MMR Alice including base, sensors and radio module.

2.2 Radio Communication and User Interface

The robot gathers data about its environment and communicate it to an host PC. One radio module is mounted on the robot and a second radio module is mounted on a "translator" board which is connected to the serial port of the computer (see Figure 2).
For testing purposes, it exists also a tiny wire connection instead of the radio modules. This permit to avoid any radio communication error and also supply the robot with energy from an external source. Even if the 4 wires (GND, VCC, TX and RX) are only 70 µm in diameter, they interfere and disturb the movement of the robot which weight only 8 g.
The radio module is composed of a receiver chip and a transmitter chip each with a separate antenna to simplify the electronics. The communication is half-duplex and follow a simple protocol with 1 start bit, 8 data bits, parity and 1 stop bit, thus error detection is possible. It is to notice that manchester encoding (0->01; 1->10) is necessary to ensure proper functionality of the radio chips but a modified version (0->00; 1->10) was implemented to decrease the power consumption during transmission which otherwise could be huge (up to 30 mW).
At the next higher level the communication protocol consider the robot as a slave with a memory where the master can write or read. It is therefore up to the Matlab program on the PC

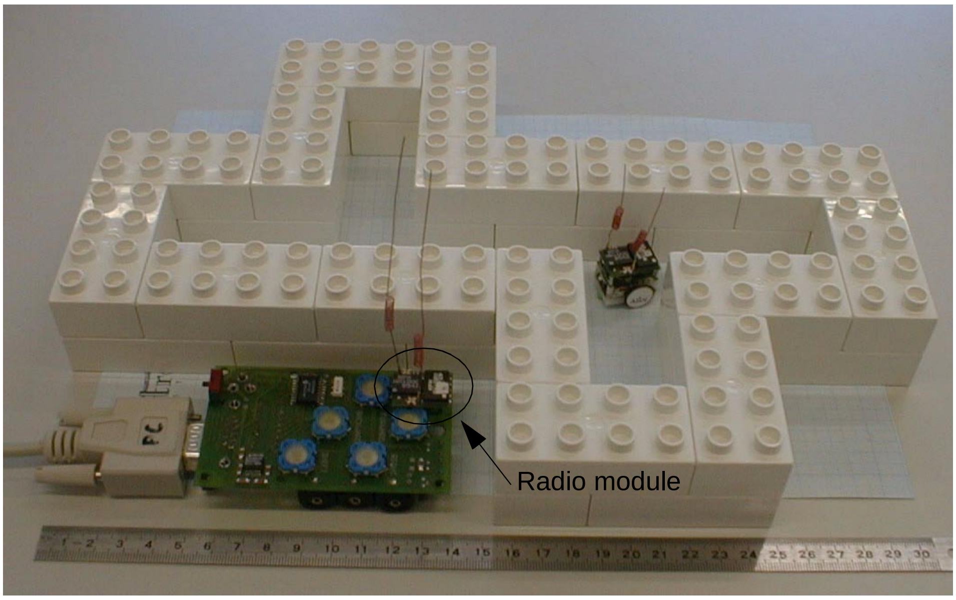

Figure 2. The entire setup Alice in a simple LEGO labyrinth, the "translator" equipped with a radio module and connected to the PC via a serial cable.

to read out the appropriate robot memory location with the sensor values and the motor increments or to write a special orders to the robot.

On the PC side, a set of Matlab executable *dll* files were developed to act as serial port drivers and provide the interface to the Matlab program which is responsible for reprocessing the row data and presenting that in a useful form and on a nice interface to the final user.

To better understand and to get an insight of the protocol from the Matlab programmer point of view, in the next few lines the essential code for a single odometry/sensors communication step is given:

nL(k)=readvar(com1,addrml);	*% read motor left increments*	*(1)*
nR(k)=readvar(com1,addrmr);	*% read motor right increments*	*(2)*
sensRL(k)=readvar(com1,addrs31);	*% read sensors right and left (compacted)*	*(3)*
sensFB(k)=readvar(com1,addrs24);	*% read sensors front and back (compacted)*	*(4)*

With the mentioned communication speed and protocol, this single cycle takes not less than 180 ms giving a maximum update frequency around 5 Hz. One consequence is a quite poor knowledge of the robot surroundings if this is running at maximum speed (40 mm/s) and thus the following algorithms have to cope with this.

3 Environment and Sensor Modeling

The sensors which are available and practicable for MMRs such as Alice impose severe limitations when reliable information for navigation is required. This is valid for both, interoperceptive sensors like odometry and for exteroperceptive sensors like range or intensity finders. This section presents how theses sensors are modeled and how their information is integrated into the environment model.

3.1 Odometry

For MMRs, and in contrast to big mobile robots, non-systematic odometry errors stemming

from uneven floors, wheel slippage or external forces appear at least in the same order of magnitude as systematic errors. This in combination with unreliable exteroperceptive sensors makes navigation a particularly difficult task. Alice does not have encoders for closed-loop displacement information from the wheels. Instead, the stepper motors allow to measure the number of steps in a open-loop manner. Clearly, this has the disadvantage that an external force blocking a wheel such that the motor looses steps can not be noticed and appears as wheel slippage to the localization system.
Using an arc approximation for each time step, assuming no external perturbations and a smooth path from the last pose, the kinematic model is then

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{2R_{Rob}}, \quad \Delta d = \frac{\Delta s_r + \Delta s_l}{2} \tag{5}$$

$$\Delta x = \Delta d \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right), \quad \Delta y = \Delta d \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right), \quad \theta(k) = \theta(k-1) + \Delta\theta \tag{6}$$

where:

$\Delta s_r;\Delta s_l$	Traveled distances for right and left wheel respectively
$(\Delta d;\Delta\theta)$	Path in robot local frame traveled in the last sampling interval
$(\Delta x;\Delta y)$	Path in global frame traveled in the last sampling interval
$\theta(k-1);\theta(k)$	Orientation in global frame before and after the interval respectively

3.2 Environment and IR Sensors

Common range sensors available for robots below the inch^3 are infrared reflexive proximity sensors or, for slightly bigger robot, ultrasonic proximity sensors. Both have a sensitivity region of conic shape, that is, they have a big opening angle providing only poorly directed range information. Further, the measured value depends strongly on the properties of the surface to be detected. This is especially true for IR sensors with an opening angle of up to 60 degrees. All

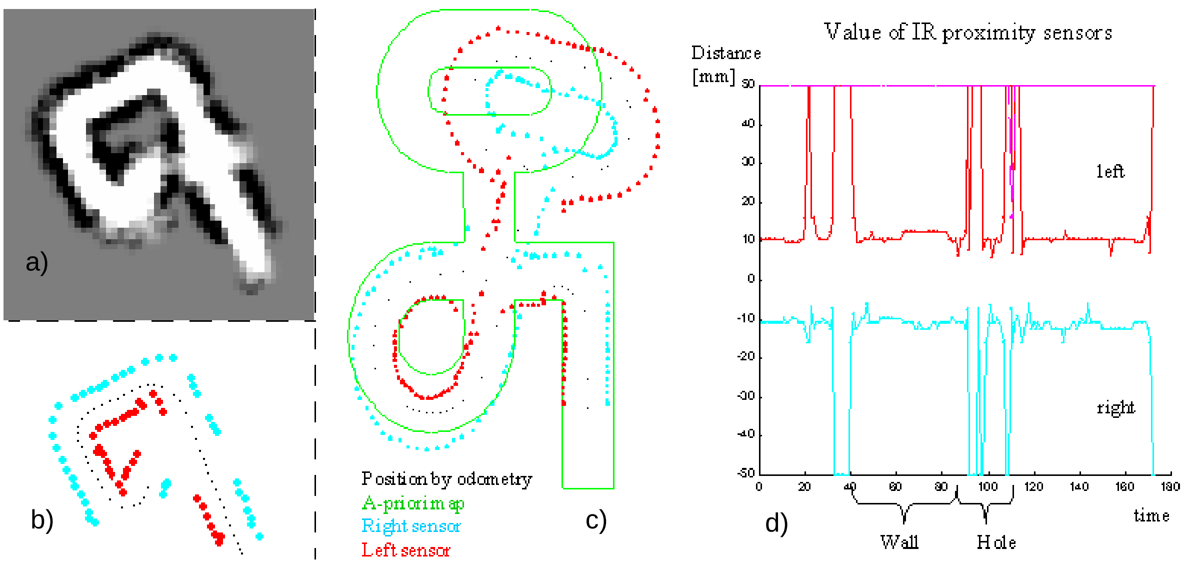

Figure 3. Proximity sensors readings in simple labyrinths: a) sensors modelled with occupancy grid method (not used for navigation), b) simple model on a straight line, c) Odometry deviation over time, d) typical values during crossing passage.

this may also have advantages (less sensors needed) but usually increases uncertainties and sensor model complexity.
In view of the these limitations there are two different models which appear suitable to integrate sensory information into local maps: *raw data* where a measure lies on a straight line in the sensor's view direction or *occupancy grids* where the measurement is geometrically distributed on a occupancy grid in front of the sensor [12]. The first one might be too simple but allows typically easy processing and less computational power whereas the second one better expresses the sensor's quality (in terms of uncertainties) but usually demands more computational cost and memory for big maps. Figure 3 depicts simple results with both methods and shows typical values when driving through a crossing.
In this work, we use the raw data model since information processing (e.g. for place recognition, section 3.3) can be done with simple rules. Since recognition results will always be unreliable with this type of sensors we believe that particularly the higher level stages shall provide the required robustness. This avoids the need to develop a more complex but perhaps better recognition with the occupancy grid approach.

3.3 Place Recognition

Most of the environments we consider here are man-made and very structured. For this, Alice extracts four different topological primitives (called *places*) which are typical for these environments: Single connection situation (dead-end, *I*), two connection situation (right- and left-sided *L*), three connection situation (*T*-crossing) and four connection situation (*X*-crossing). The extraction algorithm searches for jumps in the raw sensor readings or significant orientation changes to detect the start of a crossing. After the crossing, when the measurements are stable again, four characteristic values are compared: the mean distance value of the left, front and right sensor, and the orientation difference occurred during the intersection. Each primitive exhibits a characteristic combination of these values even though big variances occur. These places define locally unique regions which serve as points for localization. This is explained in the next section.

4 Local and Global Localization

The kind of sensory information which is available for MMRs makes metric navigation difficult. Metric navigation explicitly represents and estimates the vehicle position x, y and orientation θ in a global or local reference frame. It relies typically on precise sensory information and good models for sensors and actuators. A topology-based approach for navigation is less model-based and maintains qualitative information without the need for high precision. The robot pose is represented with respect to some locations in the environment and allows typically less accurate and intuitive formulations of the robot position: e.g. 'close to a crossing' or 'in a dead-end'. In the case of mobile mini robots, the topological approach to navigation appears to be a natural choice. The burden to accurately estimate x, y, θ with such unreliable sensors is a compelling argument for this decision.
Our approach to navigation is very similar to the one in [13] where a consistent framework is proposed allowing a robot to topologically navigate between places with a library of simple motion behaviors. In the case of Alice, these behaviors are: obstacle avoidance, wall-following left and wall following right. In this work, we additionally incorporate rough metric information in two forms: firstly we determine the robot pose with odometry and secondly we snap the

orientation θ to 0,90,180 and 270 degrees. This assumption is a limitation to a certain environment type but is still compatible with the application scenarios we consider. Using this property, odometry can be corrected to an uncritical extent and raw data can be transformed into a global reference frame with satisfying precision. The combination of the topological framework and this type of metric information yields our hybrid, topological-metric approach.

4.1 Local Localization

With local localization (also called position tracking) we refer to robot pose estimation in known environment when the previous pose is approximately known. The a priori map for localization is a simple list. Each element of the list corresponds to one of the $\{I, L, T, X\}$-places and carries their metric position. Equipped with the a priori map, the place recognition ability and the behaviors for place-to-place navigation, topological local localization is straightforward: Each time the robot traverses and recognizes a distinctive place, it searches for list elements with the identical type. These are the candidate elements. Without metric information it would be hard to uniquely determine the robot position given that there are more than one place of each type in the environment. With the topological-metric approach we can use imprecise metric information from odometry and choose the element among the candidates which is metrically closest. This element delivers the new position of the robot (Figure 4).

4.2 Global Localization

Global localization is the task of finding the robot pose in known environment without knowledge on the pose (e.g. robot is lost). For global localization the a priori map is extended with true topological information: The map is not a simple list anymore but a graph with nodes and edges. The nodes have the same meaning as the list elements before (places of type $\{I, L, T, X\}$) whereas the edges denote traversable connections between the places. In the global localization experiment, the robot navigates from its *unknown* start point randomly with the obstacle avoidance behavior. The places it traverses and recognizes are stored forming a sequence of symbols from the alphabet $\{I, L, T, X\}$. A search algorithm then tries to match the symbol sequence in the a priori known map. Multiple position hypotheses are maintained. As soon as the sequence becomes globally unique, the robot is re-localized. The matching

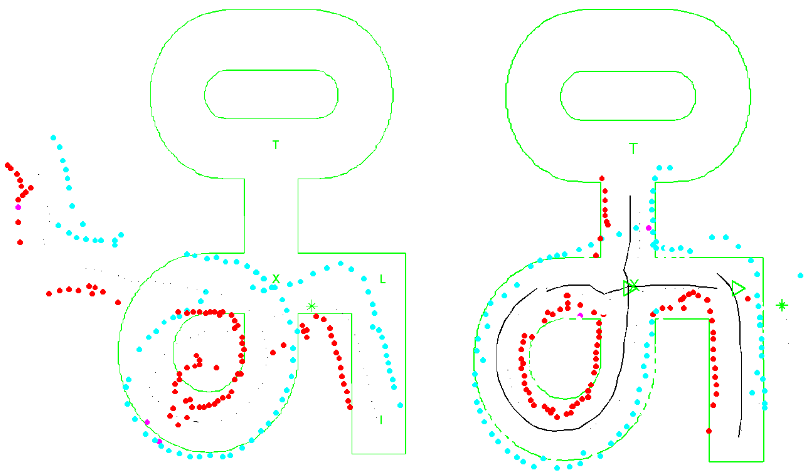

Figure 4. The effect of localization: without (left) and with (right). The points in light gray (blue) are raw range readings from the right sensor, points in dark gray (red) from the left sensor. The trace at the right displays the corrected robot trajectory. Jumps in the trace depict the localization corrections.

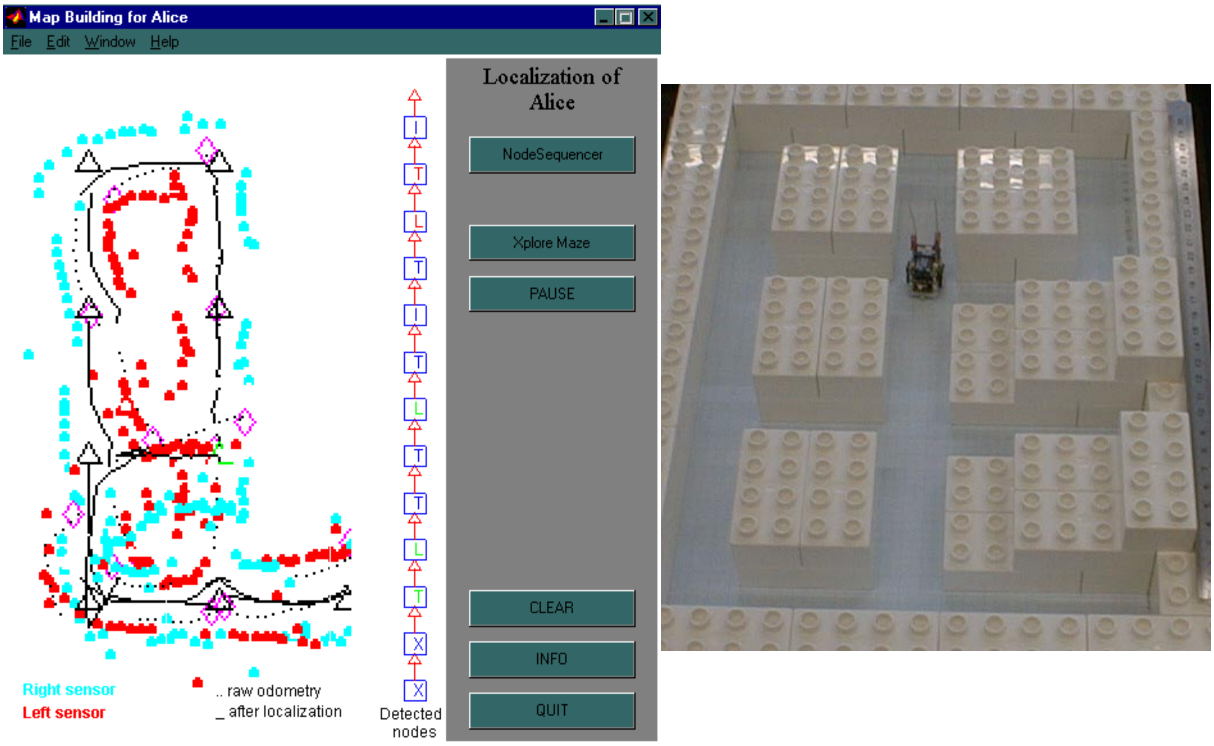

Figure 5. Global localization in the a priori known environment visible in the picture nearby and depicted in Figure 6. Corrected recognizable locations are marked by a triangle. The robot started from an unknown position. The vertical chain shows the sequence of the last thirteen detected places.

algorithm allows wildcard symbols in the sequence as well. This is of great importance since false place detection can occur due to the mentioned variability in the recognition process. In such a case, the matching stage is able to eliminate symbol sequences which are impossible in the environment. Thus, false detections can not only be recognized but also auto-corrected, yielding a high degree of robustness for localization. The corresponding Matlab interfaces are shown in Figure 5 and Figure 6.

5 Map building

Clearly, being able to automatically build maps is a very desirable ability of a mobile robot since a priori maps can be difficult to obtain. However, map building with mobile mini robots is a challenge, since, as mentioned, sensors for MMRs are usually of very low quality. Making open-loop maps (i.e. pretending that odometry is true) is therefore to be excluded. So the problem has to be addressed how re-visited places can be recognized and how sensory information can be properly aligned.

For map building, Alice detects openings with the same algorithm as for the extraction of topological primitives (section 3.3). Exploration is started from an unknown position. Unexplored openings are stored on a stack and processed with a backtracking technique. Exploration is finished when all open connections have been examined.

During the construction process, metric information is again incorporated. It turns out to do an important job for recognizing already visited places. Especially the perpendicularity assumption yields a good orientation estimate even in the absence of a map. This is important in order to correctly align the raw data and to determine the metric position of the $\{I, L, T, X\}$-places. The

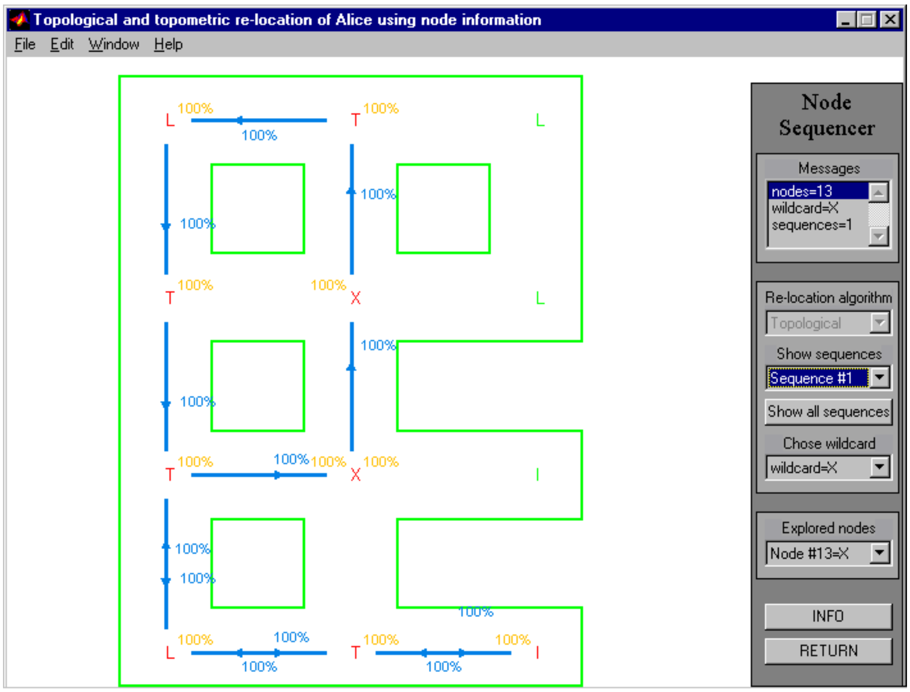

Figure 6. The environment and the path found for the symbol sequence gained during navigation of Figure 5. Started from an unknown position, the robot is successfully localized.

resulting maps (Figure 7) contain the raw data in a global reference frame and a graph representation of the environment topology.

6 Conclusion and outlook

These mini robot navigation experiments were conducted successfully with the Alice robot in simple structured labyrinths as small as possible (3 cm). The experiments demonstrate that local localization, global localization and map building is feasible with MMRs in structured

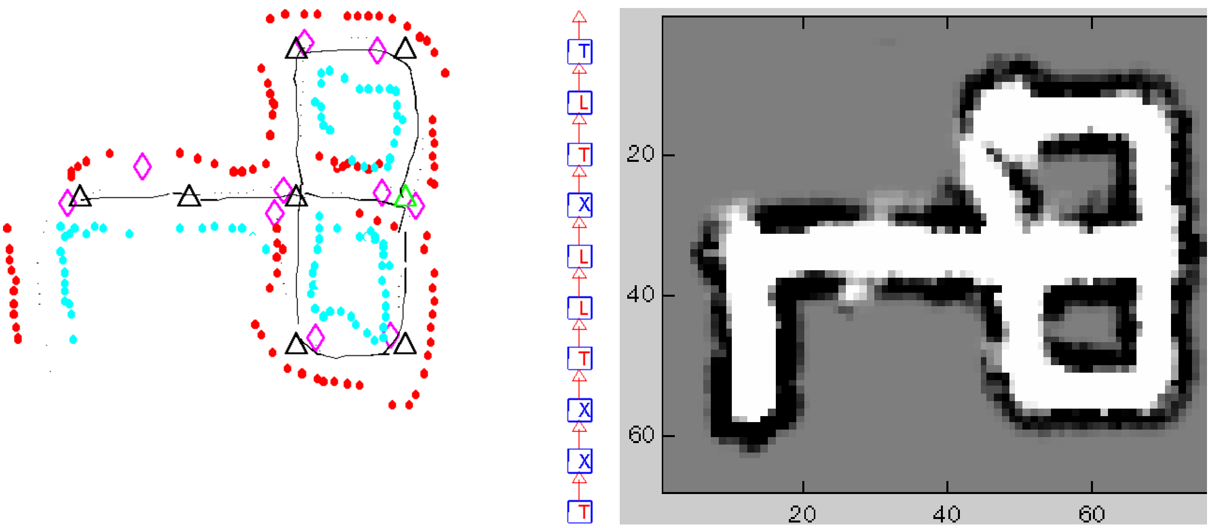

Figure 7. Maps resulting from the same exploration with odometry correction. On the left each point denotes a sensor measurement and the line is the corrected path. On the right the occupancy grid. White denotes free space, black the walls and gray the unexplored space (Not used for map building).

environments. This in spite of a typically unreliable odometry and very undirected and noisy range information. The results have been achieved with a hybrid topological-metric navigation approach using locally unique places supported by rough metric information.

Of course there are many limitation to robots in this size and Alice is surely not an exception. Many of these limitations will be overcome in the next years, encouraged by technological improvements. New solutions for better sensors like small low-power cameras are already coming out on the market and chip integration promises almost miracles. Very useful would be, for example, an integrated triangulation sensor which could provide real distance information. Another motivating point is the activity in low-power high-speed radio communication with standards like bluetooth at frequencies which permit, for instance, smaller antennas.

However, two problems will remain for a longer period of time: power limitation and imprecise odometry. The first one is inherent to the size and the second one is, among other reasons, due to the downscaling effect of the robot mass compared with its characteristic length. The mass of small robots has few impact to its movements, so more slippage occurs and already weak external forces can have a drastic effect. Therefore, poor odometry has to be defeated by something else. As this paper demonstrated, simple and structured environments can help to work around this problem. Another interesting way to explore is multi-robot navigation/ exploration. On the other hand there is still enough place for smart and new solutions, maybe even mechanical ones.

Acknowledgements. Special thanks to the students who made during their semester project a significant work on this topic: Cédric Glauser, Edouard Meylan and Goran Savatic.

References

[1] **LAMI-EPFL**, Switzerland, The Microrobots Jemmy and Inchy, *diwww.epfl.ch/lami/mirobots/1cubes.html*

[2] **J. McLurkin**, Using Cooperative Robots for Explosive Ordonance Disposal, MIT AILab, 1996 *http://www.ai.mit.edu/projects/ants/papers.html*

[3] **H. Ishihara and T. Fukuda**, Miniaturized Autonomous Robot, *SPIE* vol.3202, pp. 191-6, 1998

[4] **L.E. Navarro-Serment** et al., Modularity in Small Distributed Robots, *SPIE vol. 3839*, pp. 297-306, 1999.

[5] **Sandia National Laboratories**, New Release - Mini-robot research,USA, January 2001, *http://www.sandia.gov/media/NewsRel/NR2001/minirobot.htm*

[6] **International Micro Robot Maze Contest**, Nagoya, Japan, *http://www.mein.nagoya-u.ac.jp/maze*

[7] **F. Mondada, E. Franzi, P. Ienne**, Mobile robot miniaturization: A tool for investigation in control algorithms, *Proc. of the 3rd International Sym. On Experimental Robotics*, pp. 501-513, 1993

[8] **R. Siegwart,** et al., Guiding Mobile Robots through the Web, Workshop Proc. of IROS 98, Victoria, Canada, pp. 5-10, October 1998.

[9] **P. Saucy**, Conception d'un environnement réparti pour le contrôle de robots mobiles distants, chapter on Khepera Positioning System, *Thesis 2142, EPFL*, 2000.

[10] **L.E. Navarro-Serment, C.J.J. Paredis, P.K. Khosla**, A Beacon System for the Localization of Distributed Robotic Teams, *Proc. of the Int. Conference on Field and Service Robotics*, Pittsburgh, August 1999.

[11] **G. Caprari, P. Balmer, R. Piguet, R. Siegwart**, The Autonomous Micro Robot ALICE: A platform for Scientific and Commercial Applications, *MHS'98*, pp 231-5, Japan, 1998.

[12] **H.P. Moravec, A. Elfes**, High Resolution Maps from Wide Angle Sonar, *Proceedings of the IEEE conf. on Robotics and Automation*, pp 116-121, Washington, D.C., 1985.

[13] **B.J. Kuipers, Y.T. Byun**, A Robust, Qualitative Approach to a Spatial Learning Mobile Robot, *Proceedings of the SPIE, Sensor Fusion: Spatial Reasoning and Scene Interpretation*, Vol. 1003, 1988.