SLAM and Map Merging

A. León, R. Barea, L.M. Bergasa, E. López, M. Ocaña and D. Schleicher

Abstract—**This paper presents a multi-robot mapping and localization system. Learning maps and efficient exploration of an unknown environment is a fundamental problem in mobile robotics usually called SLAM (simultaneous localization and mapping problem). Our approach involves a team of mobile robots that uses Scan-Matching and Fast-SLAM techniques based on scan-laser and odometry information for mapping large environments. The single maps generated for each robot are more accurate than the maps generated only by odometry. When a robot detects another, it sends its processed map and the master robot generates a very accurate global map. This method cuts down the global map building time. Some experimental results and conclusions are presented.**

Index Terms— **Multi-robot SLAM, scan-mathcing, fast-slam, rao-blackwellised particle filter.**

I. Introduction

LEARNING maps and efficient exploration of unknown environments is a fundamental problem in mobile robotics. This problem is usually called SLAM (simultaneous localization and mapping problem) [1, 2, 3, 4, 5, 6], which includes estimating the robot's position on the map and building up a map using the sensory input and the estimated robot´s pose.

The problem of exploration of an unknown environment has been extensively studied, first using single robot systems with a variety of sensors and later using teams of robots. The first multi-robot exploration systems implementations were simple extensions of the single robot implementations. Multiple robot systems are more complex than other distributed systems because they have to deal with a real environment, which is more difficult to model since it is dynamic, unpredictable, noisy, etc. Therefore, the extension to multiple robots systems brings several new challenges and difficulties [7][8]: robot coordination, integration of information collected by different robots into a consistent map and dealing with limited communication.

Multirobot exploration systems are usually classified as centralized and decentralized. Centralized systems obtain solutions close to the optimal but they are computationally intensive and its good functioning depends on the central module. On the other hand, decentralized systems are flexible and robust, but frequently achieve significantly suboptimal solutions. Therefore, the difficulty of the coordination task strongly depends on the knowledge of the robots. If the robots know their relative locations and share a map of the area they explored so far, then effective coordination can be achieved by guiding the robots into different, nonoverlapping areas of the environment [9], [10], [11]. However, if the robots do not know their relative locations, then it is far less obvious how to effectively coordinate them, since the robots do not share a common map or reference frame [7].

Map merging task consists on building a consistent model of an environment with data collected from different robots. If the initial locations of the robots are known, map merging is a rather straightforward extension of a single robot mapping [12], [13], [14]. If robots do not know their relative locations, it is more difficult, since it is not clear how and where the robots' traces should be connected.

One of the hardest problems in robotic mapping is the loop closure [15]. When a robot navigates throughout a large cycle in the environment, it must face the hard data association problem of connecting correctly the data to its own map under large position errors. To scale to large-scale environments, one option consists on transform sequences of laser range-scans into odometry measurements using range-scan registration techniques [16], which reduces the well-known particle deprivation problem [17][18].

Rao-Blackwellized particle filters have been introduced as effective means to solve the simultaneous localization and mapping (SLAM) problem. This approach uses a particle filter in which each particle carries an individual map of the environment [19], [20]. The main problem of the Rao-Blackwellized approaches is their complexity, measured in terms of the number of particles required to build an accurate map. To solve this, Hahnel et al [15] combine Rao-Blackwellized particle filtering and scan matching with an improved motion model that reduces the number of required particles. Grissetti et al. [21] present an adaptive technique to reduce the number of particles in a Rao-Blackwellized particle filter for learning grid maps, they propose an approach to compute an accurate proposal distribution taking into account not only the movement of the robot but also the most recent observation. This drastic decrease the uncertainty about the robot's pose in the prediction step of the filter.

This paper presents a comparison of different grid-based SLAM algorithms over some application examples. A technique for merging maps from several robots is also described. The objective is to build up a highly accurate map of an unknown environment with unknown initial

R. Barea is with the Electronics Department, University of Alcalá, Spain (e-mail: barea@depeca.uah.es).

L.M. Bergasa is with the Electronics Department, University of Alcalá, Spain (e-mail: bergasa@depeca.uah.es).

E. López Guillén is with the Electronics Department, University of Alcalá, Spain (e-mail: elena@depeca.uah.es).

M. Ocaña is with the Electronics Department, University of Alcalá, Spain (e-mail: mocana@depeca.uah.es).

position of robots using Rao-Blackwellized particle filtering and scan matching. Finally, some experimental results of our proposal and the main conclusions will be drawn and commented.

II. PERFORMANCE COMPARISON OF GRID-BASED SLAM ALGORITMS

This section makes a comparison of different SLAM algorithms. To do that, firstly, we will study the basic of the SLAM techniques then, we will compare them over some application examples and finally we will present some conclusions about its performance.

A. *Scan-Matching SLAM*

The scan matching algorithm that we have implemented is an extension of the approach presented in [16], where the problem of concurrent mapping and localization can be treated as a maximum likelihood estimation problem, in which the aim is to find the most likely map given the data:

Let *m* be a map. At time *t*, the map is written:

$$m_t = \{\langle o_t, x_t \rangle\}_{t=0,\dots,T} \qquad (1)$$

where o_T denotes a laser scan and x_T its pose, and t is time index.

The goal of mapping is to find the most likely map given the data, that is:

$$\underset{\mathrm{m}}{\operatorname{argmax}}\, P(m \mid d_t) \qquad (2)$$

where data $\mathrm{d_t}$ is a sequence of laser range measurements and odometry readings:

$$\mathrm{d_t} = \{\mathrm{o_0, a_0, o_1, a_1, :::, o_T}\} \qquad (3)$$

where *o* denotes an observation (laser range scan), *a* denotes an odometry reading, and *t* are time indexes.

The map likelihood function *P(m | d_t)* can be transformed into the following product [18]:

$$P(m \mid d_t) = \eta \cdot P(m) \int \dots \int \prod_{t=0}^{T} P(o_T \mid m, x_t) \cdot$$
$$\cdot \prod_{t=0}^{T-1} P(x_{t+1} \mid a_t, x_t) dx1 \dots dx_t \qquad (4)$$

where η is a normalizer and *P(m)* is prior over maps which, if assumed to be uniform, can safely be omitted. Thus, the map likelihood is a function of two terms, the motion model, $P(x_{t+1} \mid a_t, x_t)$, and the perceptual model, $P(o_t \mid m_t, x_t)$. If stationarity is assumed (i.e., neither model depends on the time index t), the time index can be omitted and instead write $P(x \mid a, x')$, for the motion model and $P(o \mid m, x)$ for the perceptual model.

B. *Grid-based Fast-SLAM*

This algorithm adapts the Fast-SLAM algorithm to occupancy grid maps. This gives us a volumetric representation of the environment that does not require any predefined landmark and it can therefore model arbitrary types of environments. The pseudo-code for grid-based Fast-SLAM [22] in each iteration is the following:

```
For i=0 to M do
    $x_t^i$ = motion model $(a_t, x_{t-1})$
    $w_t^i$ = model map $(o_t, x_{t-1}^i, m_{t-1}^i)$
    $m_t^i$ = update_map $(o_t, x_{t-1}^i, m_{t-1}^i)$
    $\tilde{X}_t = \tilde{X}_t + \langle x_t^i, m_t^i, w_t^i \rangle$
endfor

For i=0 to M do
    Draw i with probability $\alpha \cdot w_t^i$
    Add $\langle x_t^i, m_t^i \rangle$ to $X_t$
Endfor
```

C. *Rao-Blackwellized mapping*

The key idea of the Rao-Blackwellized particle filter [21] for SLAM is to estimate a posterior $p(x_{1:t} \mid o_{1:t}, a_{0:t})$ about potential trajectories $x_{1:t}$ of the robot given its observations $o_{1:t}$ and its odometry measurements $a_{0:t}$ and to use this afterwards to compute a posterior over maps and trajectories:

$$p(x_{1:t}, m | o_{1:t}, a_{1:t}) = p(m | x_{1:t}, o_{1:t}) \cdot p(x_{1:t} | o_{1:t}, a_{1:t}) \quad (5)$$

This can be done efficiently, since the posterior over maps $p(m | x_{1:t}, o_{1:t})$ can be computed analytically given the knowledge of $x_{1:t}$ and $o_{1:t}$.

To estimate the posterior $p(x_{1:t} | o_{1:t}, a_{1:t})$ over the potential trajectories Rao-Blackwellized mapping uses a particle filter in which an individual map is associated to every sample. Each map is built given the observations $o_{1:t}$ and the trajectory $a_{1:t}$ represented by the corresponding particle. The trajectory of the robot evolves according to the robot motion and for this reason the proposal distribution is chosen to be equivalent to the probabilistic odometry motion model.

One of the most common particle filtering algorithms is the Sampling Importance Resampling (SIR) filter. A Rao-Blackwellized SIR filter for mapping incrementally processes the observations and the odometry readings as they are available. This is done by updating a set of samples representing the posterior about the map and the trajectory of the vehicle.

This is done by performing the following four steps:

- Sampling: The next generation of particles x_t^i is obtained from the current generation x_{t-1}^i by sampling from a proposal distribution $\pi(x_t | o_{1:t}, a_{1:t})$.

- Importance Weighting: An individual importance weight w^i is assigned to each particle, according to

$$w^i = \frac{p(x_t^i | o_{1:t}, a_{1:t})}{\pi(x_t^i | o_{1:t}, a_{1:t})} \qquad (6)$$

The weights w^i account for the fact that the proposal distribution B in general is not equal to the true distribution of successor states.

- Resampling: Particles with a low importance weight *w* are typically replaced by samples with a high weight. This step is necessary since only a finite number of particles are used to approximate a continuous distribution. Furthermore, resampling allows to apply a particle filter in situations in which the true distribution differs from the proposal.
- Map Estimating: for each pose sample x_t^i, the corresponding map estimate m_t^i is computed based on the trajectory and the history of observations according to $p(m_t^i | x_{1:t}^i, o_{1:t}^i)$.

D. Performance comparison

Firstly, we are going to compare the different SLAM algorithms using a manufactured rectangular map (figure 1). The objective of this initial test is to study the performance of these algorithms in environments with large corridors without points of interest (points where the symmetry of other types of information is conducive to accurate location).

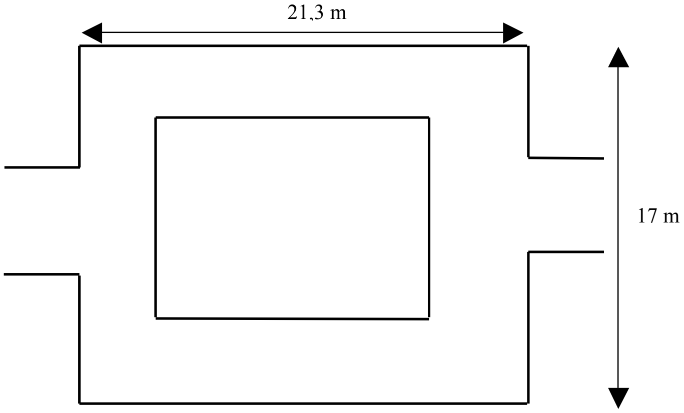


Fig. 1. Manufactured map.

Figures 2 and 3 show the map obtained after one and three map laps, respectively, using the scan-matching SLAM algorithm.

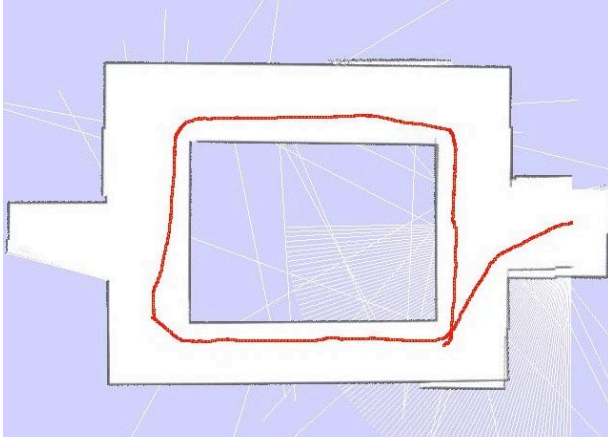


Fig. 2. Initial map using scan-matching SLAM.

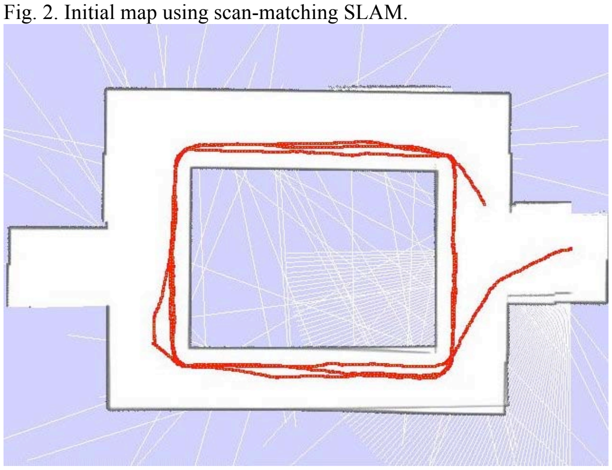


Fig. 3. Scan-matching map (3 laps).

Figures 4 and 5 show the map obtained after one and three map laps, respectively, using the grid-based Fast-SLAM algorithm.

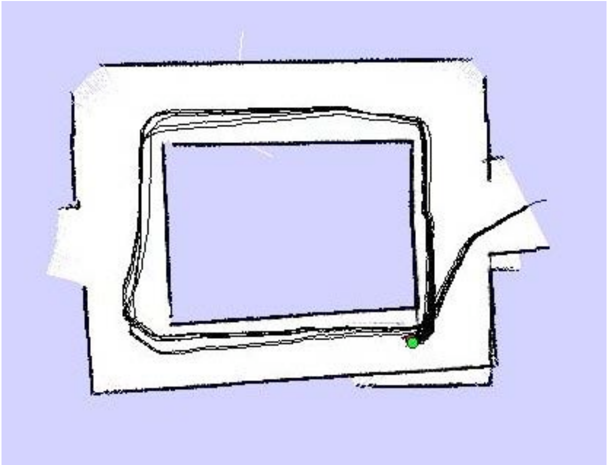


Fig. 4. Initial map using grid-based Fast-SLAM.

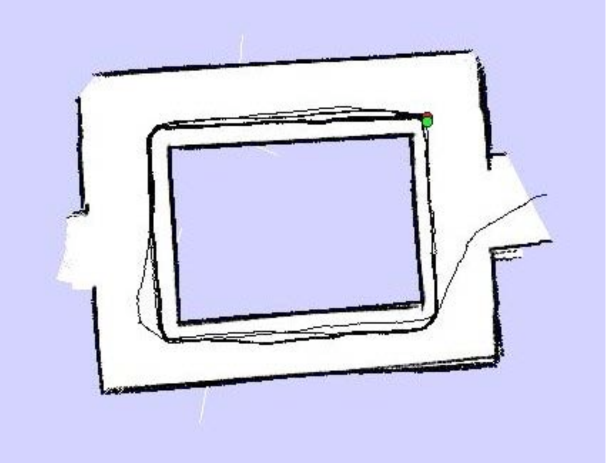


Fig. 5. Grid-based FastSlam map (3 laps).

Figures 6 and 7 show the map obtained after one and three map laps, respectively, using the Rao-Blackwellized particle filter for SLAM.

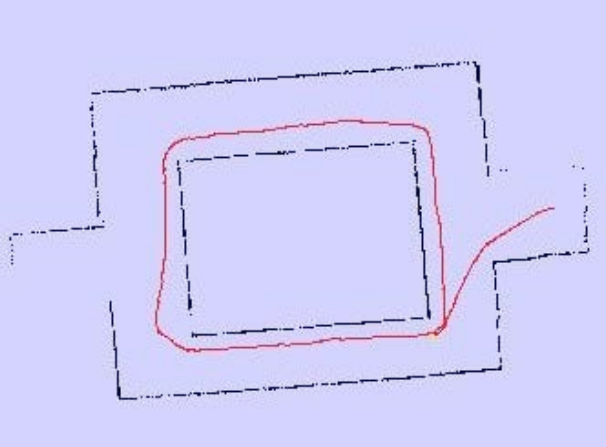


Fig. 6. Initial map using Rao-Blackwellized particle filter for SLAM.

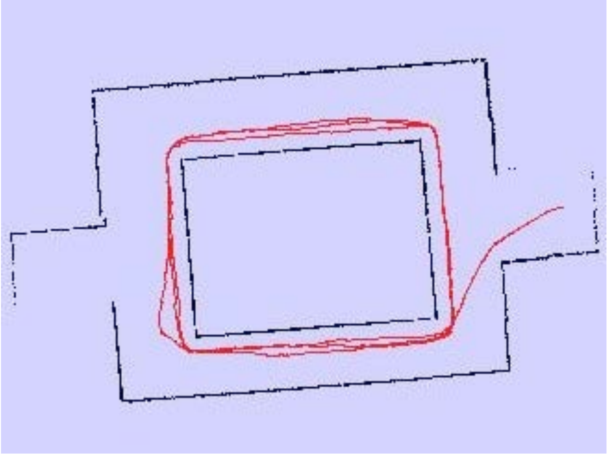


Fig. 7. Rao-Blackwellized Slam map (3 laps).

All cases under study show that errors decrease and results improve with the number of laps. This stands to reason, since the system progressively solves the mapped errors detected in the first lap.

Figure 8 shows the three maps obtained, superimposed over the original map. The red unbroken line shows the original map, the green dot-and-dash line (-·-) traces out the map obtained using scan-matching, the orange dashed line (--) shows the map generated using grid-based Fast-SLAM while the blue dotted line (···) shows the map generated using Rao-Blackwellized mapping.

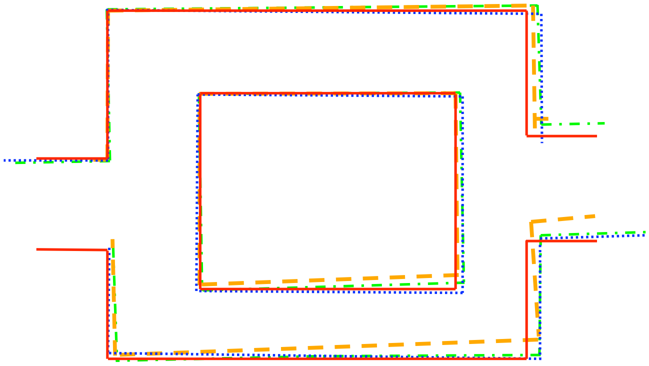


Fig. 8. Mapping errors (3 laps).

Table I shows the dimensional errors detected using the three algorithms. After three laps these errors decrease very slowly and then improvements are very low.

TABLE I
DIMENSIONAL ERROR

	Real	Scan-Matching	Grid-based Fast-SLAM	Rao-Blackwellized
Length	21.3 m	22.3	21.7	22.4
Error_length		4.6%	1.8%	5.1%
Heigh	17 m	17.1	16.8	16.7
Error_heigh		0.5%	1.1%	1.7%

These results show that the detected errors are small and negligible in small-dimension environments. The best results would seem to be obtained by the grid-based Fast-SLAM algorithm, but the map geometry in fact shows that the generated map in this case is the worst of the three.

One of the problems that appear in environments of this type (especially for the case of scan-matching) is that if there is not outstanding information in the corridors (points of interest) the results record the overall corridor length smaller than the real one. This is because the maximum likelihood estimation method is used, in which the aim is to find the most likely map given the data. This occurs especially when we work in real environments and is due mainly to the environment-detection limitations of the sensor used. In our case this sensor is a laser SICK-LMS200 and in the corridors when the end is not detected, there are many points where the laser measurement is the same or very similar. The maximum likelihood estimation algorithm therefore generates significant estimation errors.

The next objective is to study a more complex environment. Figure 9 shows a manufactured map from the Electronics Department of University of Alcala (corridor 3 and 4). Figure 10 shows the map obtained using odometry and laser data without any correction. The robot starts in Lab_1, goes to Lab_2 and returns to Lab_1.

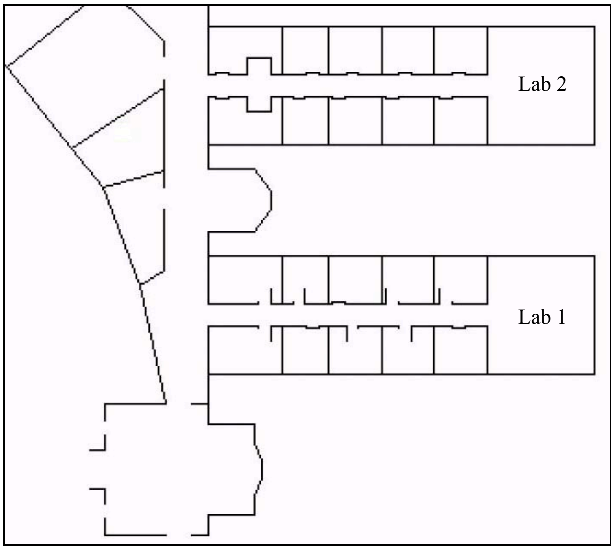


Fig. 9. Department of Electronics map (Corridor 3 and 4).

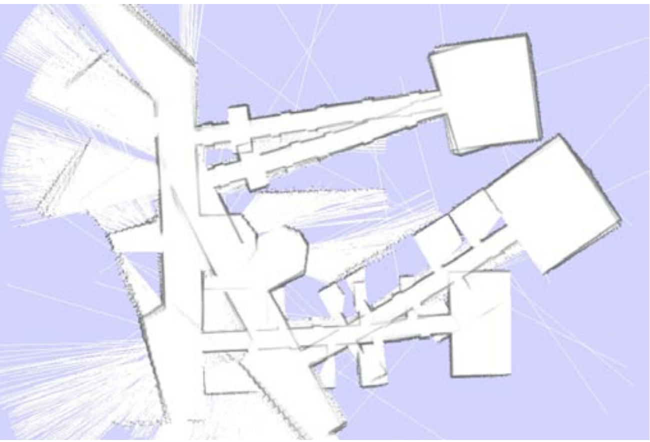


Fig. 10. Map obtained using odometry and laser data without correction.

Figures 11, 12 and 13 show the map obtained using scan-matching SLAM, grid-Fast-SLAM and grid-based Rao-Blackwellized SLAM, respectively. All algorithms solve existing errors using odometry and laser data, but a certain error exists in the orientation of the corridors.

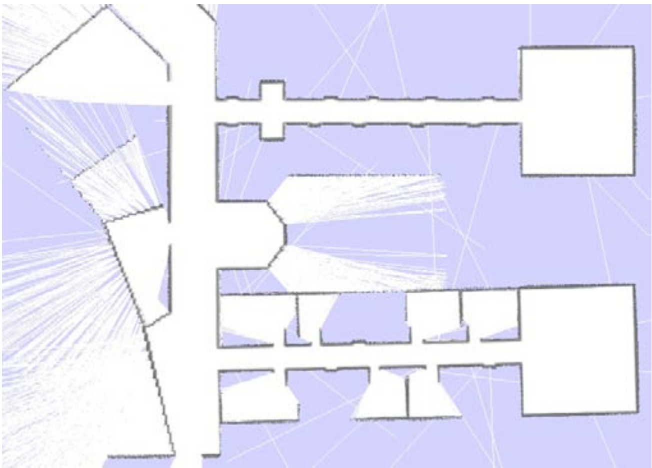


Fig. 11. Map obtained scan-matching SLAM

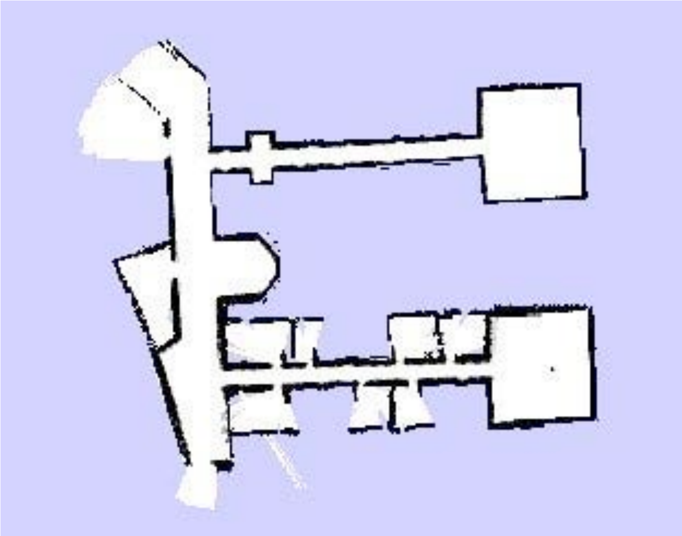


Fig. 12. Map with Fast-SLAM.

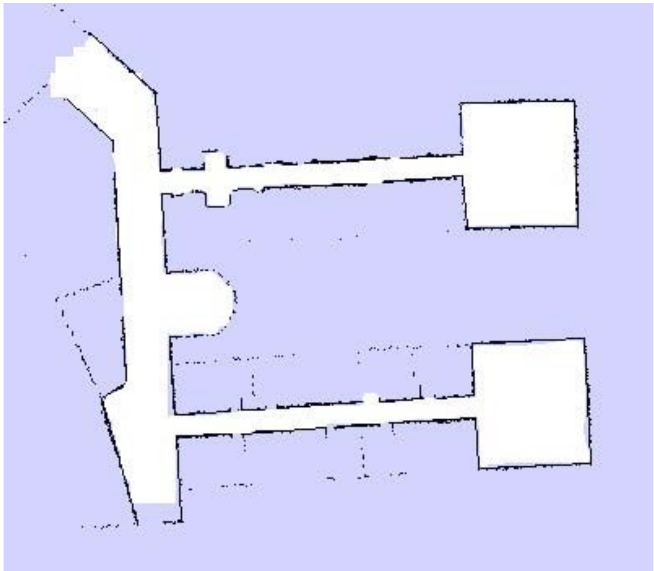


Fig. 13. Map with grid-based Rao-Blackwellized SLAM.

Another experiment to determine the errors introduced by these algorithms involves mapping a larger real environment as is the whole third floor of the Polytechnic building, as can be seen in Fig. 14.

The aim in this case is to compare results using scan-matching and grid-based Rao-Blackwellized SLAM. To do that a lap around the building will be carried out and the map generation errors will be studied. Figures 15 and 16 show the results obtained from one complete lap of the building. The start and the end point is a research laboratory in office 34. Robot trajectory has been superimposed on the real map to compare map-building errors.

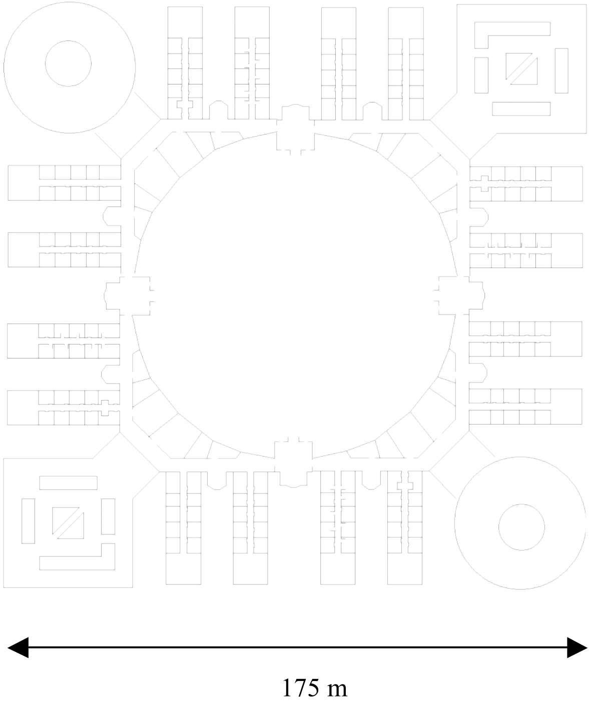


Fig. 14. 3ª Floor – Politechnic School.

Both cases show that significant position estimation errors occur after one lap of the building. Figure 15 (scan-matching) shows a position error of about 2 meters, while Figure 16 (Rao-Blackwellized SLAM) gives a 6 meter error. It should be bore in mind here that more than 400 meters have been traveled without any loop closing or point of interest that might allow correction of the existing map. Simulations suggest that results improve after several laps around the building. The results obtained can therefore be considered satisfactory.

III. Architecture

A. Robots

Four robotic platforms have been developed based on PeopleBot, pioneer DX and pioneer AT robots of ActivMedia Robotics [23] (see figure 17). Their architecture comprises four large modules: environment perception, navigation, human-machine interface and high-level services. The perception module is equipped with encoders, bumpers, sonar ring, laser sensor and a vision system based on a PTZ (pan-tilt-zoom) color camera connected to a frame grabber. The human-machine interface is composed of loudspeakers, microphone, a tactile screen, the same PTZ camera used in the perception module, and a wireless Ethernet link. The high-level services block controls the rest of the modules and includes several tasks of tele-assistance, tele-monitoring, providing, reminding and social interaction [24].

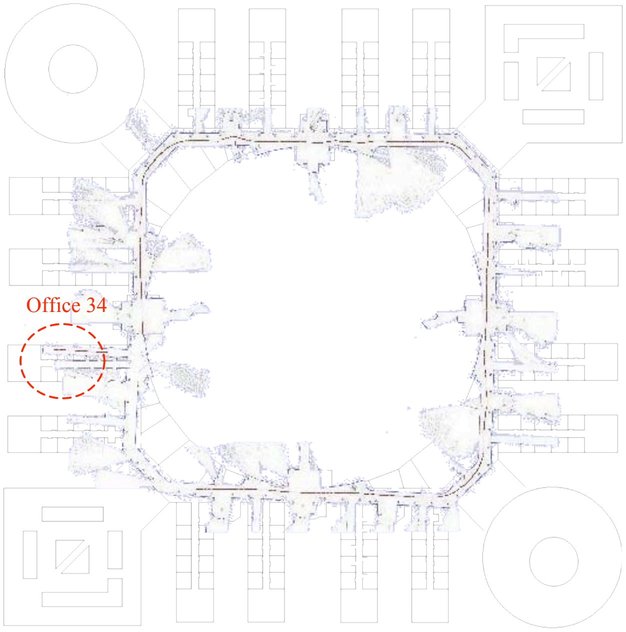


Fig. 15. Scan-matching SLAM map. Comparison of real and obtained map.

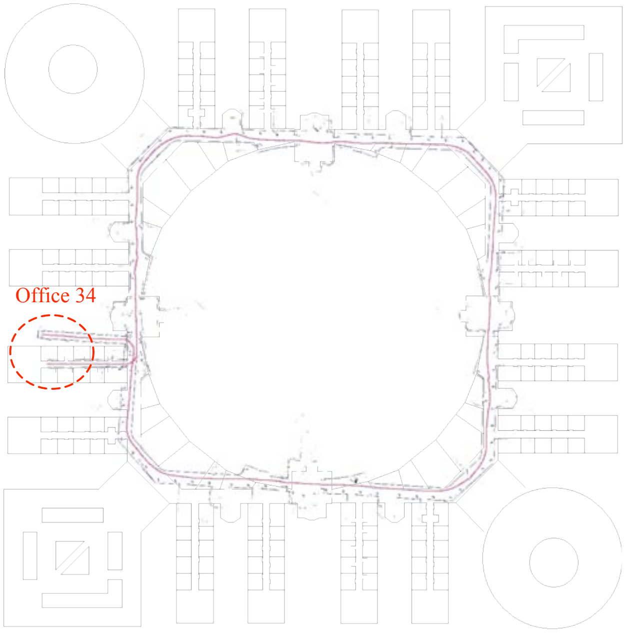
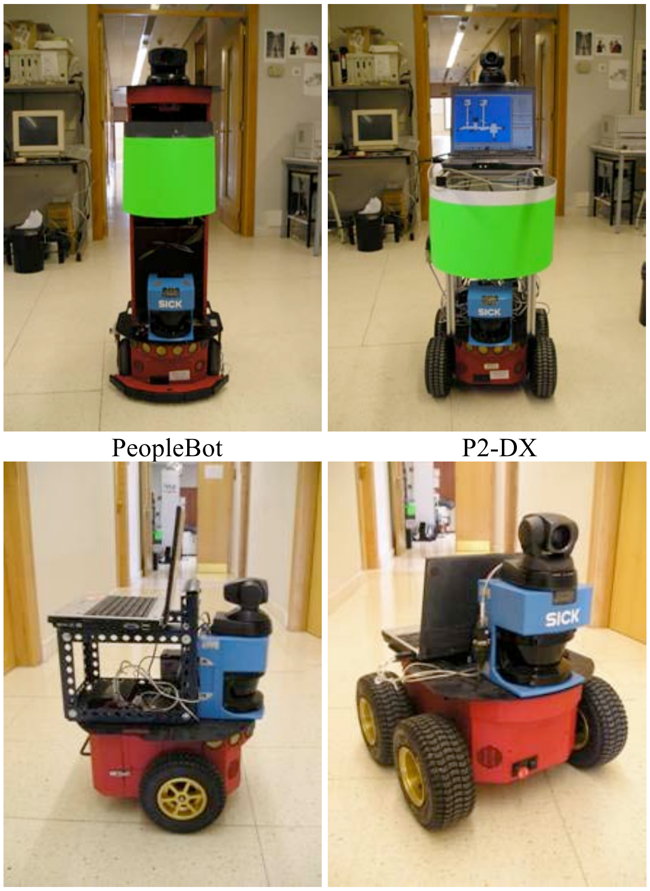


Fig. 16. Grid-based Rao-Blackwellized SLAM map. Comparison of real and obtained map.



PeopleBot P2-DX

P3-DX P3-AT

Fig. 17. Robots.

B. Navigation module

The navigation module combines information from the perception module for carrying out different tasks. The core of this module is CARMEN (Carnegie Mellon Robot Navigation Toolkit) [25] which is an open-source collection of software for mobile robot control. CARMEN is modular software designed to provide basic navigation primitives including: base and sensor control, obstacle avoidance, localization, path planning, people-tracking, and mapping.

This source has been modified to implement multi-robot localization, because CARMEN permits only single-robot working; there is also a different initial distribution for studying the robot localization (based on Montecarlo localization). This source implements the motion, perception and detection models. A virtual simulator has also been developed for testing the detection model using visual information and the localization process.

C. Detection model

Each robot has the ability to sense each others. The detection model describes the probability that robot n is at location x, given that robot m is at location x´ and perceives robot n with measurement r_m.

To determine the relative location of other robots, our approach combines visual information obtained from an onboard camera with proximity information coming from a laser range-finder. Camera images are used to detect other robots and, together with laser range-finder scans, determine the relative position of the detected robot and its distance.

The robots are marked by a unique and colored marker to facilitate their recognition (green cylinder). The marker can therefore be detected regardless of the robot's orientation.

To find robots in the images obtained from a camera, our system first filters the image by employing local color histograms (HIS space color). A basic segmentation algorithm, based on the histogram, is then employed to search for the marker's color. If it is found, this implies that a robot is present in the image.

Once a robot has been detected, its size and position in the image are processed and a laser scan is made for calculating the relative position of one robot respecting the other (see figures 18 to 20). This multi-sensor technique has been proven to be robust in practical tests. Currently, images are analyzed approximately at 20 fps. This framerate is enough for this application.

When a robot detects the other one, a detection model is generated (based on a Gaussian function) representing the probability that the detected robot is at this point. This detection model carries out the adjustment of the particles by means of Collaborative Monte Carlo localization.

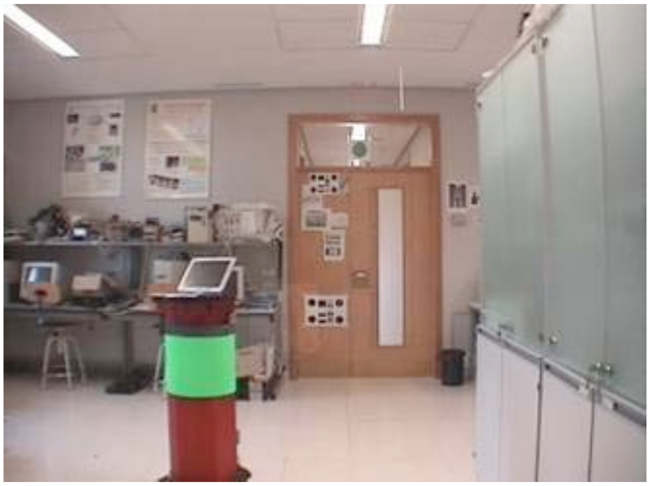


Fig. 18. Original image.

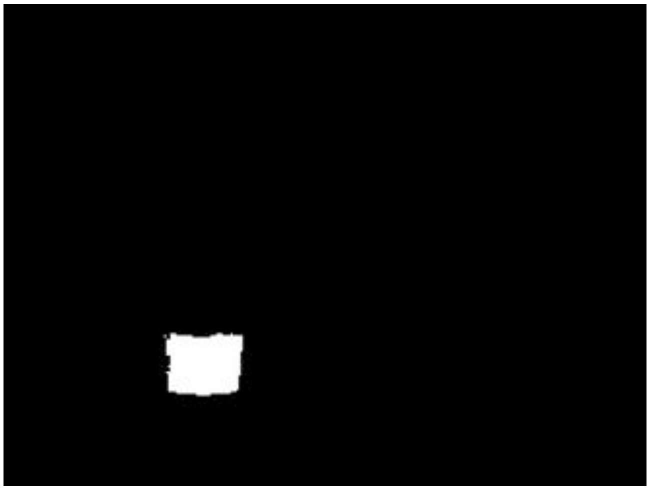


Fig. 19. Processed image.

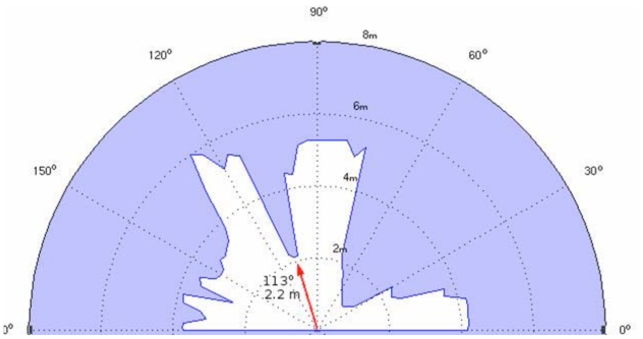


Fig. 20. Laser scan. Robot detected.

IV. Multi-robot Monte Carlo localization

Monte Carlo Localization has been widely studied in [6,14]. MCL is a recursive Bayes filter that estimates the posterior distribution of robot poses conditioned on sensor data.

The key idea of Bayes filtering is to estimate a probability density over the state space *x* conditioned on the data. This posterior is typically called the *belief* and is denoted:

$$Bel(x_t) = p\left(x_t \middle| \; o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, o_0\right) \quad (7)$$

Where x_t is the vector state at time *t*, *o* denotes *observations* (*perceptual data* such as laser range or vision measurements) and *a* represents *actions* (*odometry data* which carry information about robot motion).

Bayes filters estimate the belief recursively. The initial belief characterizes the initial knowledge about the system state. In the absence of such knowledge, it is typically initialized by a uniform distribution over the state space. In mobile robot localization, a uniform initial distribution corresponds to the global localization problem, where the initial robot pose is unknown.

To derive a recursive update equation, (7) can be transformed by Bayes rule to:

$$Bel(x_t) = \eta \cdot p(o_t|x_t, a_{t-1}, \ldots, o_0)\, p(x_t|x_{t-1}, a_{t-1}, \ldots, o_0) \quad (8)$$

$$\eta = p(o_t|a_{t-1}, \ldots, o_0)^{-1} \quad (9)$$

Bayes filters assume that the environment is Markov, that is, past and future data are (conditionally) independent if one knows the current state. The Markov assumption implies:

$$p(o_t|x_t, a_{t-1}, \ldots, o_0) = p(o_t|x_t) \quad (10)$$

$$p(x_t|x_{t-1}, a_{t-1}, \ldots, o_0) = p(x_t|x_{t-1}, a_{t-1}) \quad (11)$$

Therefore, the belief can be denoted by:

$$Bel(x_t) = \eta \cdot p(o_t|x_t) \cdot \int p(x_t|x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (12)$$

Where $p(o_t|x_t)$ is called *perceptual model* and $p(x_t|x_{t-1},a_{t-1})$ represents the *motion model*.

The key idea of multi-robot localization is to integrate measurements taken at different platforms, so that each robot can benefit from data gathered by other robots than itself. Therefore, when a robot *n* is detected by robot *m* it is necessary to introduce the detection model according with data obtained r_m in (12). In the absence of detections, the Markov localization works independently for each robot. A summary of the multi-robot Markov localization algorithm is:

- Initialize the belief $Bel_n(x)$ according with initial data (typically uniform distribution).
- If the robot *n* receives an observation on (new sensory input) o_n, it is applied the perception model::

$$Bel_n(x) = \eta \cdot p(o_n \mid x) \cdot Bel_n(x) \quad (13)$$

- If the robot n do some action a_n (receives a new odometry reading), It is applied the motion model:

$$Bel_n(x') = \eta \cdot \int_x p(x' | x, a_n) \cdot Bel_n(x) \cdot dx \quad (14)$$

- And finally, if the robot n is detected by the m-th robot it is applied the detection model:

$$Bel_n(x') = \eta \cdot Bel_n(x) \int_x p(x_n = x' \mid x_m = x, r_m) \cdot Bel_m(x) \cdot dx \quad (15)$$

The idea of MCL is to represent the *belief* by a set of *m* weighted samples distributed according to *Bel(x)*:

$$Bel(x_t) \approx \{x^i, w^i\}_{i=1,\ldots,m} \quad (16)$$

Where x^i is a *sample* of the random variable *x* (pose) and w^i is called *importance factor* and represents the importance of each sample. The set of samples, thus, define a discrete probability function that approximates the continuous belief *Bel(x)*.

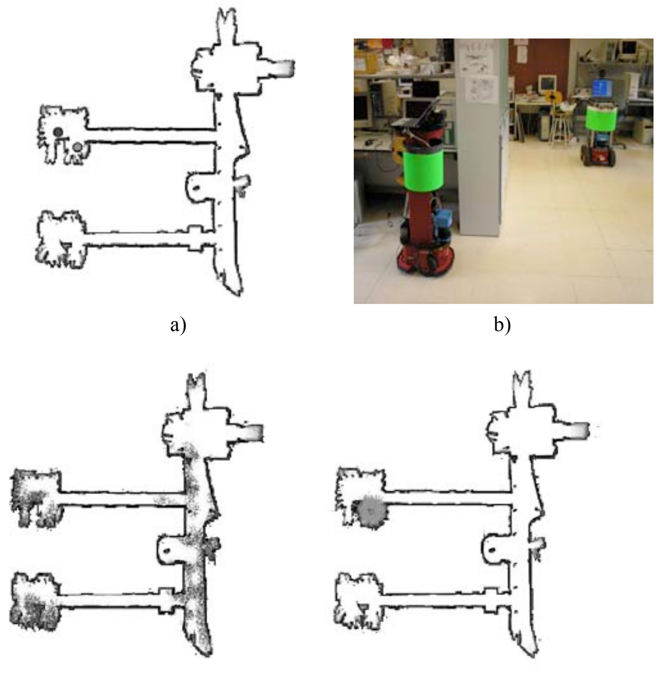
The initial set of samples represents the initial knowledge $Bel(x_o)$ about the state of the dynamical system. For instance, in global mobile robot localization, the initial belief is a set of poses drawn according to a uniform distribution over the robot's universe, annotated by the uniform importance factor 1/m. If the initial pose is known up to some small margin of error, $Bel(x_o)$ may be initialized by samples drawn from a narrow Gaussian centered on the correct pose.

The recursive update is carried out in three steps:

- Sample x^i_{t-1}– $Bel(x_{t-1})$. Each such particle x^i_{t-1} is distributed according to the belief distribution $Bel(x_{t-1})$.
- Sample x^i_{t-1}– $p(x_t|x^i_{t-1},a_{t-1})$. In this case, x^i_t is distributed according to the product distribution $p(x_t|x_{t-1},a_{t-1})\cdot Bel(x_{t-1})$.
- The importance factor is assigned to the i-th sample:

$$w^i = 0 \cdot p(o_t|x^i_t) \quad (17)$$

The following example shows how collaborative multi-robot Monte Carlo localization improves single localization [15]. Robot 1 starts out with uniform belief and Robot 2 with Gaussian belief. Figs. 21a.b show the robots' initial position and Figs. 21c.d show the initial particle distributions.

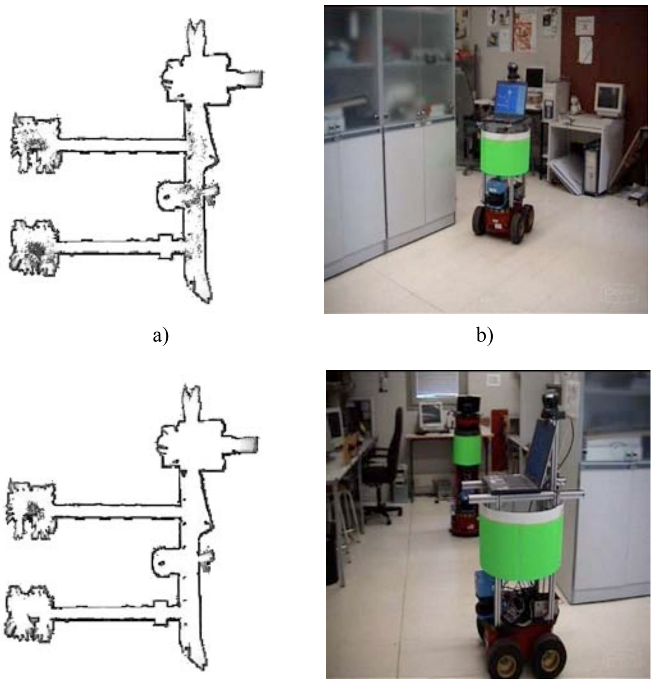


a) b)

c) d)

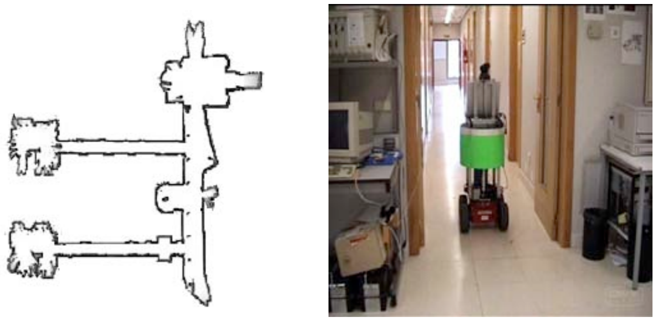
Fig. 21. Initial distributions.

If Robot 1 wanders across the top horizontal corridor (Fig. 22a), when Robot 2 detects Robot 1 (Fig. 22b), the detection model is sent to Robot 1 and this updates its belief distribution. Robot 1 is therefore well-located before reaching the corridor (Fig. 22c).



a) b)

c) d)



e) f)

Fig. 22. Collaborative multi-robot Monte Carlo localization.

V. MULTIROBOT MAP MERGING

This section describes how to build a map from data obtained by multiple robots. We are currently working with scan-match [16], grid-based Fast-SLAM [15] and grid-based SLAM with Rao-Blackwellized particle filter [21] for map merging; comparing the results. To do so, we have modified some codes and parameters in algorithms (GMapping and GridSLAM) obtained from OpenSlam.org [26] which provides SLAM researchers with a platform for publishing their algorithms.

In this initial work we are focused on developing multirobot map merging using a scan-match technique. Next, a merging map example is commented on, working with a scan-match technique (see figures 23 to 27). The goal is mapping corridors 3 and 4 of the Electronics Department of the Polytechnic School. Figure 23 shows the trajectories followed for each robot and figure 24 shows the global map using CARMEN (Montecarlo localization). Robot 1 explores across corridor 4 and Robot 2 explores corridor 3; each robot builds its partial map and calculates its pose at any time. When Robot 1 meets Robot 2, Robot 2 send its map to Robot 1. Robot 1 uses the partial Robot 2 map and the detection model (Robot 2´s pose detected) to generate the global map (Fig. 27).

The main problem we run into here is the need for very precise inter-robot detection for map merging purposes. Any small orientation error would mean that the map built up from the slave robot data would be displaced vis-à-vis the master robot. We are currently working to improve inter-robot detection, making it much more precise in the interests of correct global map reconstruction.

Our final goal is to be able to map exteriors and implement other SLAM techniques for reconstructing the most trustworthy environment possible. One of the problems we are finding is that 2D laser information outdoors may be of little value in wider settings and we are therefore trying to merge laser with vision. We are thus working towards the introduction of visual information in localization and mapping process to improve SLAM's performance outdoors.

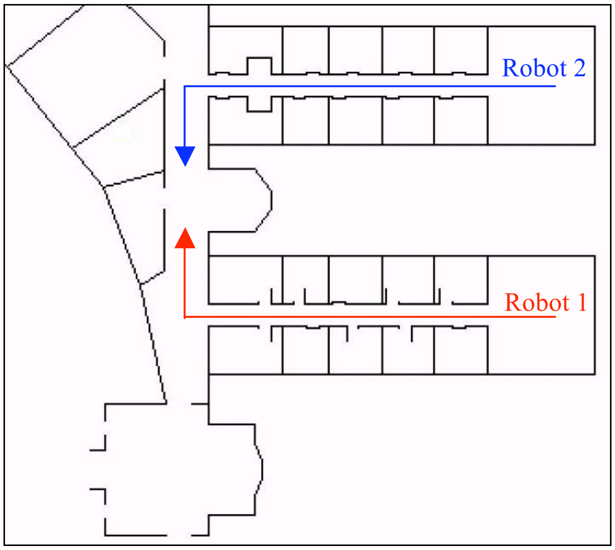


Fig. 23. Robots trajectories.

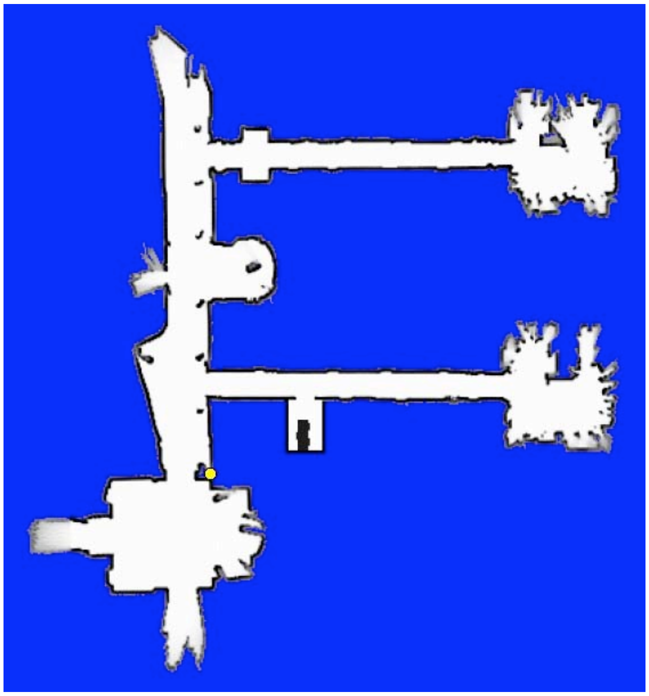


Fig. 24. Complete map obtained using CARMEN.

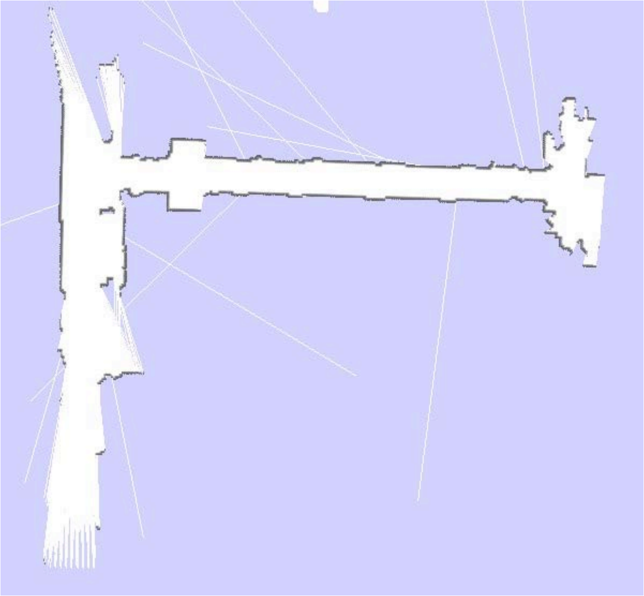


Fig. 25. Partial map built by robot 2.

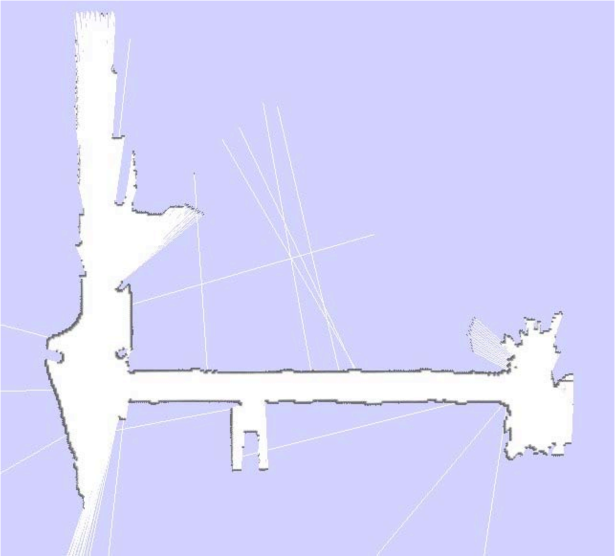


Fig. 26. Partial map built by robot 1.

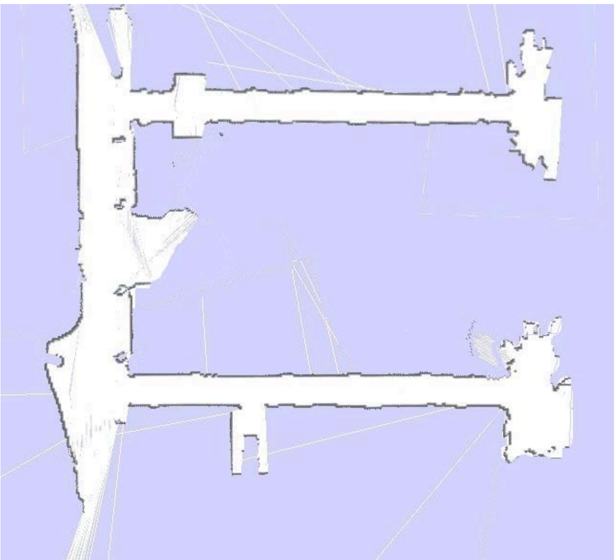


Fig. 27. Global map using scan-matching.

VI. Conclusion

We present herein initial work in mobile robot mapping. We show and comment on results using several algorithms for SLAM (scan-matching, grid-FastSLAM and grid-based SLAM with Rao-Blackwellized particle filter) for a single robot and multi-robot map merging using a scan-match technique. The results show that it is possible to use a team of robots to explore and navigate in unknown environments.

Our future work will be focused on improve performance of our proposal for indoor environments and to generalize it for outdoor environments. According to this point, the results obtained show that it is necessary to use other sources of information (3D laser, vision, etc) to obtain similar results to those obtained indoors.

Acknowledgment

The authors would like to express their gratitude to the Comunidad de Madrid and the University of Alcalá for their support through the projects RoboCity2030 (CAM-S-0505/DPI/000176) and SLAM-MULEX (CCG07-UAH/DPI-1736).

References

[1] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. IEEE Transactions on Robotics and Automation, 15(5):948–953, 1999.

[2] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localization and map building (SLAM) problem. In ICRA'2000 Workshop on Mobile Robot Navigation and Mapping, 2000.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping (SLAM): Part I", in IEEE Robotics & Automation Magazine. 2006.

[4] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II", in IEEE Robotics & Automation Magazine. 2006.

[5] M. Montemerlo, S. Thrun D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), 2003.

[6] H. Chang,; C.Lee,; Y. Hu,; Y. Lu, "Multi-robot SLAM with topological/metric maps", in Intelligent Robots and Systems, 2007.

[7] D. Fox, J. Ko. K. Konolige, B. Limketkai, D. Schulz, B. Stewart, "Distributed Multirobot Exploration and Mapping", in Proceedings of the IEEE, Volume 94, Issue 7, pp:1325 – 1339, July 2006.

[8] Parker, L.E. "Current state of the art in distributed robot systems", in *Dist. Auton. Syst*., L.E. Parker, G. Bekey and J. Barhen (Eds.), vol. 4, pp. 3–12. Springer-Verlag. 2000.

[9] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration", in Proc. IEEE Int. Conf. Robotics Automation (ICRA), 2000.

[10] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration", in IEEE Trans. Robot., vol. 21, no. 3, pp. 376–386, Jun. 2005.

[11] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping", in Proc. Nat. Conf. Artificial Intelligence (AAAI), 2000.

[12] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams", in Experimental Robotics: 9th Int. Symp., Springer Tracts in Advanced Robotics (STAR),M. Ang and O. Khatib, Eds. New York: Springer Verlag, 2005.

[13] S. Thrun, W. Burgard, D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping", in Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference, April 2000.

[14] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards multi-vehicle simultaneous localisation and mapping", in Proc. IEEE Int. Conf. Robotics Automation (ICRA), 2002.

[15] D. Hahnel, W. Burgard, D. Fox, D.S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements", in Intelligent Robots and Systems, 2003.

[16] D. Hahnel, D. Schulz, W. Burgard, "Map building with mobile robots in populated environments", In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2002.

[17] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots", in International Journal of Robotics Research, 20(5):335–363, 2001.

[18] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/FINFENG/ TR 380, Cambridge University, Department of Engineering, 2000.

[19] K. Murphy. "Bayesian map learning in dynamic environments". In Neural Info. Proc. Systems (NIPS), 1999.

[20] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russel. "Raoblackwellized particle filtering for dynamic bayesian networks. In Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI), 2000.

[21] G. Grisettiy, G. Stachniss, W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling", in Robotics and Automation, 2005.

[22] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics". Cambridge, MA: MIT Press, 2005.

[23] R. Barea, LM. Bergasa, E. López, D. Schleicher, M. Ocaña and A. León. "Real Patient monitoring in health care working with robotic assistants", in Proc. Of the IEEE Internacional Symposium on Intelligent Signal Processing (WISP). Pp:711-716. October 2007.

[24] R. Barea, E. López, L.M. Bergasa, S. Álvarez and M. Ocaña. "Detection Model in Collaborative Multi-Robot Monte Carlo Localization". IEEE Workshop on Distributed Intelligent Systems (DIS'06), pp. 49-54, 2006.

[25] Carnegie Mellon Robot Navigation Toolkit. http://www.cs.cmu.edu/~carmen/index.html

[26] www.openslam.org.