Multi-robot 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 unknown environment is a fundamental problem in mobile robotics usually called SLAM (simultaneous localization and mapping problem). Our approach uses a team of mobile robots which uses scan-matching and fast-slam techniques for mapping. The single maps generates for each robot are more accurate than map generated by odometry data. When a robot detects another, send its processed map and the master robot can generates a global map very accurate. This way, time necessary to build the global map is reduced.**

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

I. Introduction

LEARNING maps and efficient exploration of unknown environment 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 position of the robot relative to the map and building a map using the sensory input and the estimated robot´s pose.

The problem of exploration of an unknown environment has been extensively studied, firstly using single robot systems with a variety of sensors and later using teams of robots. The first implementations of multirobot exploration systems 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]: coordination of robots, 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 are computationally intensive and have a single point of failure. 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, non-overlapping 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 frame of reference [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 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 that of loop closure [15]. When robot navigates a large cycle in the environment, it faces the hard data association of correctly connecting 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] combines 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 system for map merging from several robots. The objective is to built a highly accurate map of unknown environment with initial position of robots unknown using Rao-Blackwellized particle filtering and scan matching and compare them.

II. Scan-Matching SLAM

The algorithm to scan matching that we have implemented is an extension of the approach presented in [16], where the

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).

problem of concurrent mapping and localization can be treated as a maximum likelihood estimation problem, in which one seeks to find the most likely map given the data:

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

$$m_t = \{\langle o_T, s_t \rangle\}_{t=0,\ldots,T} \tag{1}$$

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

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

$$\underset{m}{\operatorname{argmax}}\, P(m \mid d_T) \tag{2}$$

where data d_T is a sequence of laser range measurements and odometry readings:

$$d_t = \{s_0, a_0, s_1, a_1, :::, s_T\} \tag{3}$$

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

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

$$P(m \mid d_t) = \eta \cdot P(m) \int \ldots \int \prod_{t=0}^{T} P(o_T \mid m, s_t) \cdot \cdot \prod_{t=0}^{T-1} P(s_{t+1} \mid a_t, s_t) ds1 \ldots ds_t \tag{4}$$

where *0* 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, $P(s_{t+1} \mid a_t, s_t)$, and the perceptual model, $P(o_t \mid m_t, s_t)$. If stationarity is assumed (i.e., neither model depends on the time index t), the time index can be omitted and instead write $P(s \mid a, s')$, for the motion model and $P(o \mid m, s)$ for the perceptual model.

Next, an example of performance of SLAM using scan-matching compared with simple odometry and data representation is commented:

Figure 1 shows a manufactured map from Department of Electronics of University of Alcalá (corridor 3 and 4). Figure 2 shows the map obtained using odometry and laser data without any correction and figure 3 shows the map obtained using scan-matching SLAM. It can be see that scan-matching SLAM solves the errors existing using odometry and laser data.

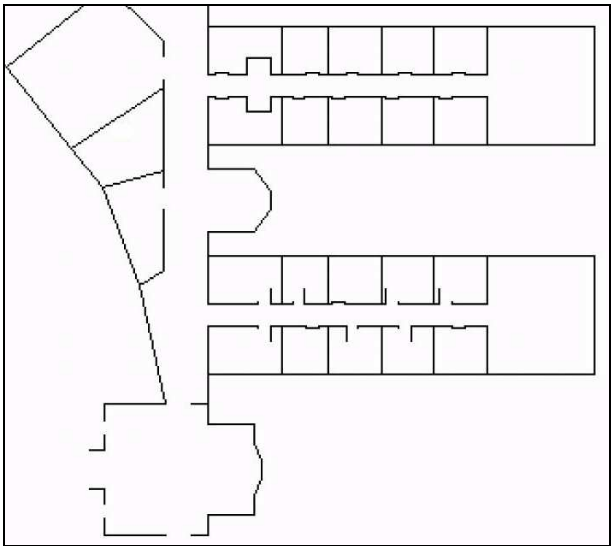

Fig. 1. Department of Electronics map (Corridor 3 and 4).

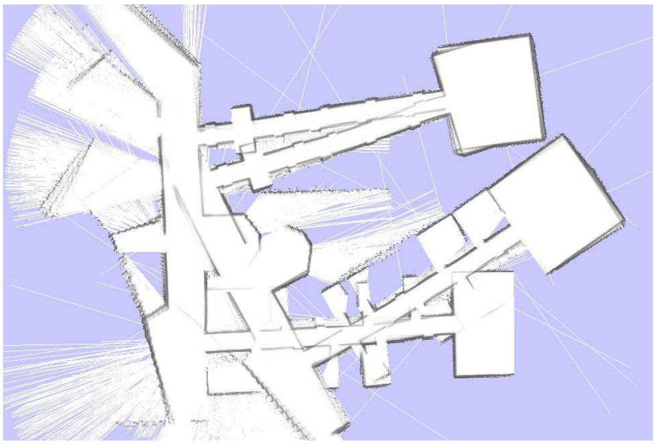

Fig. 2. Map obtained using odometry and laser data without correction.

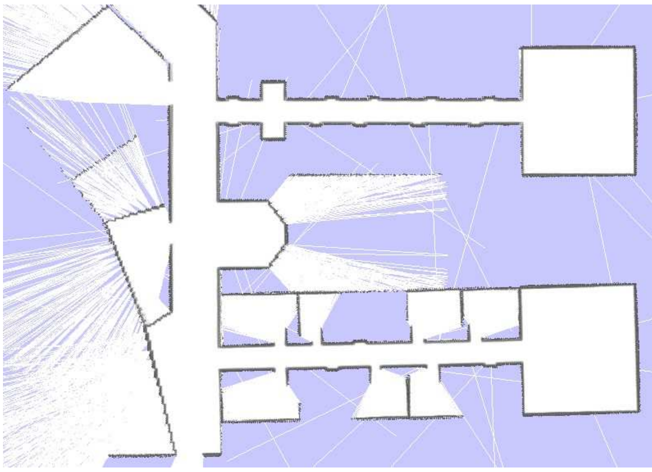

Fig. 3. Map obtained scan-matching SLAM

.

III. GRID-BASED FASTSLAM

This algorithm adapts the FastSLAM algorithm to occupancy grid maps. This way, we can obtain a volumetric representation of environment that it does not require any predefined landmark, and therefore can model arbitrary types of environments. The pseudo-code for grid-based FastSLAM [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

Figure 4 shows the map obtained using grid-fastslam.

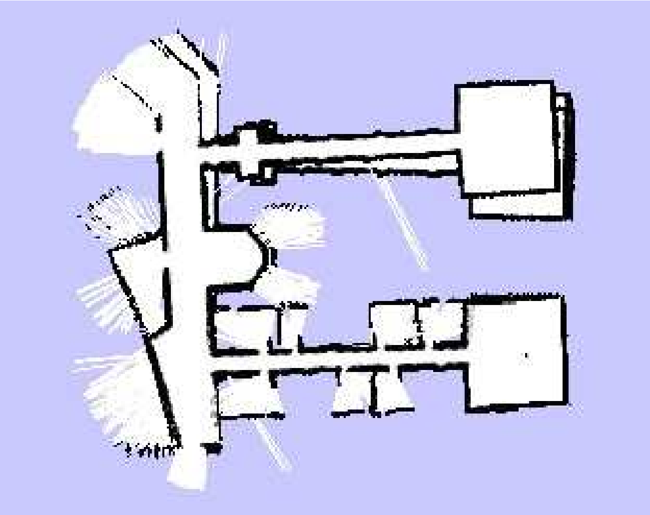

Fig. 4. Map with Fast-SLAM.

IV. 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 posterior 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

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

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 $p(m_t^i|x_{1:t}^i, o_{1:t}^i)$.

Figure 5 shows the map obtained using an improving grid-based SLAM with Rao-Blackwellized particle filters.

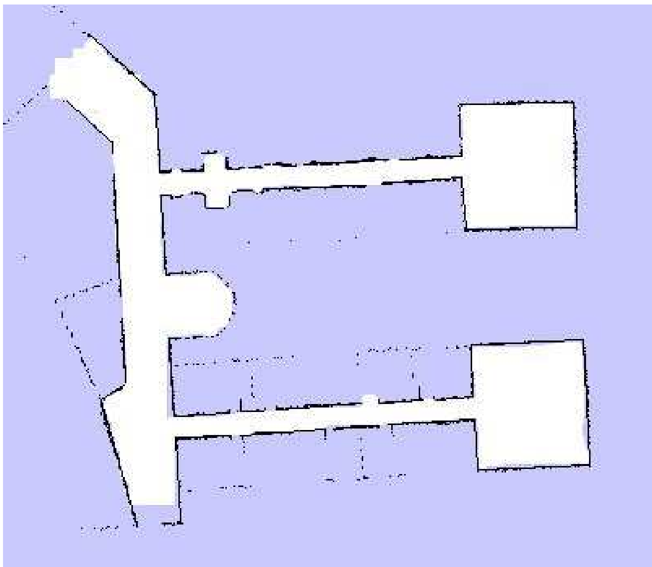

Fig. 5. Map with grid-based Rao-Blackwellized SLAM.

V. ARCHITECTURE

A. Robots

Four robotic platform have been developed based on PeopleBot, pioneer DX and pioneer AT robots of ActivMedia Robotics [23] (see figure 6). Its architecture is composed of four large modules: environment perception, navigation, human-machine interface and high-level services. The perception module is endowed 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 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].

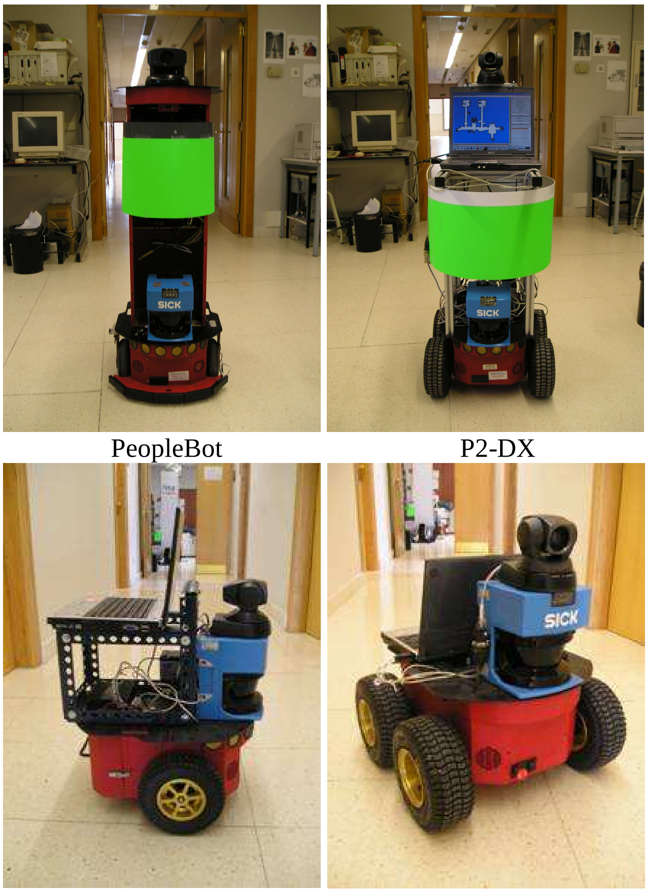

Fig. 6. Robots.

B. Navigation module

The navigation module combines information from 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 the multi-robot localization, because CARMEN only permits works with a single robot, and different initial distribution for studying the robot localization. This source implements the motion, perception and detection models. Besides, a virtual simulator has been developed for testing the detection model using visual information and the localization process.

C. Detection model

Robots must possess the ability to sense each other. 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 on-board 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 are used to determine the relative position of the detected robot and its distance.

The robots are marked by a unique and colored marker to facilitate its recognition (green cylinder). This way, the marker can be detected regardless of the robot’s orientation.

To find robots in a camera image, our system first filters the image by employing local color histograms (HIS space color). Thresholding is then employed to search for the marker’s characteristic color transition. If found, this implies that a robot is present in the image.

Once a robot has been detected, we process de object detected (size and position in the image) and the laser scan for calculating the relative location of the robot (see figures 7 to 9). This multi-sensor technique has been proved robust in practical test. Currently, images are analyzed approximately at a rate of 10Hz, this rate is sufficient for the application at hand.

When a robot detects to the other one a detection model is generated (usually type Gaussian) that it represents the probability that the detected robot is in this point. This detection model carries out the adjustment of the particles by means of Collaborative Monte Carlo localization.


Fig. 7. Original image.

Fig. 8. Processed image.

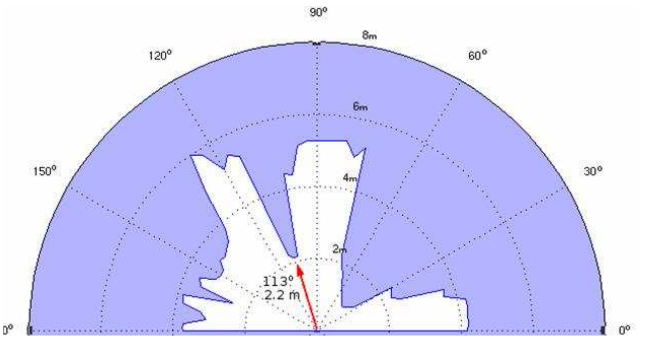

Fig. 9. Laser scan. Robot detected.

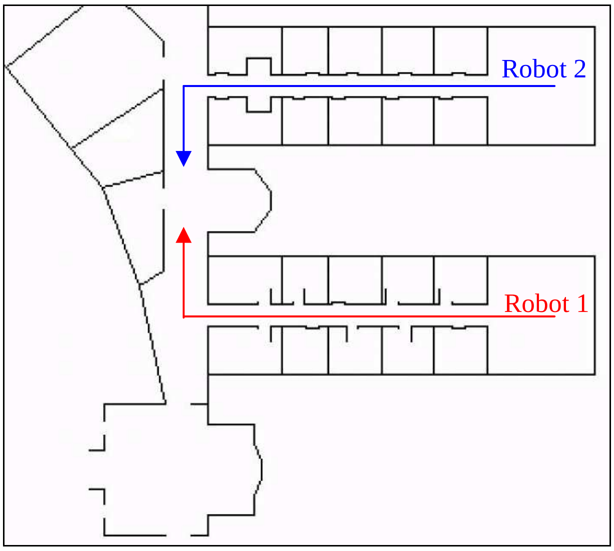

Fig. 10. Robots trajectories.

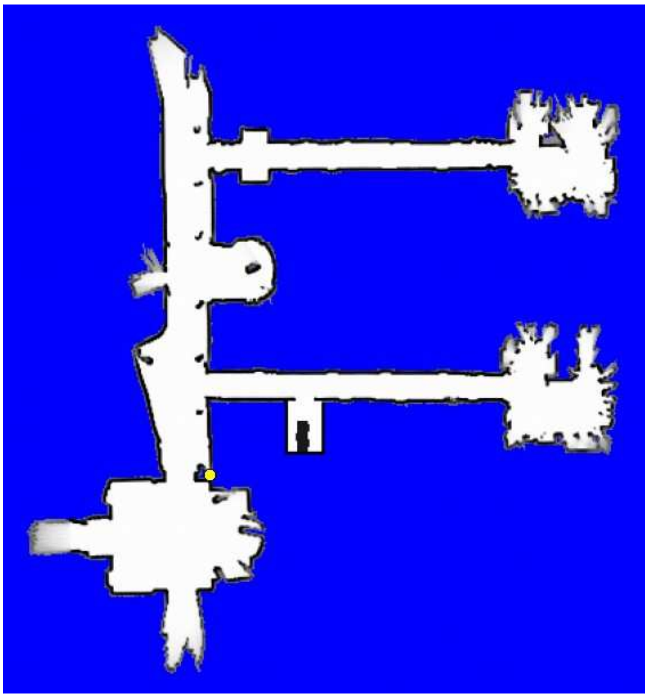

Fig. 11. Complete map obtained using CARMEN.

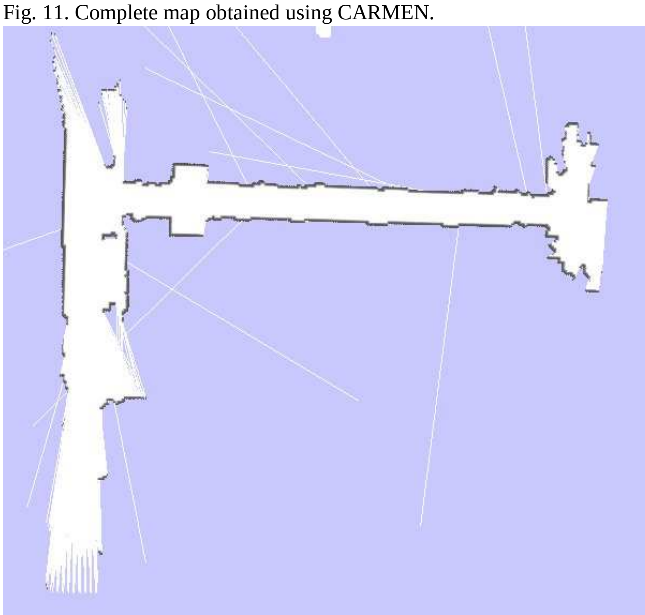

Fig. 12. Partial map built by robot 2.

VI. MULTIROBOT MAP MERGING

This section describes how to build a map from data obtained by multiple robots. Nowadays, we are working with scan-match [16], grid-based fastslam [15] and grid-based fastslam with Rao-blackwellized particle filter [21] in map merging and compared them. To do it, we have modified some code and parameters in algorithms (GMapping and GridSLAM) obtained from OpenSlam.org [26] which provides SLAM researchers a platform for publishing their algorithms.

In this initial work we focus our goal in developing a multirobot map merging using scan-match technique. Next, a merging map example is commented working with scan-match technique (see figures 10 to 14). The goal is built the corridor 3 and 4 from Department of Electronics of University of Alcalá. Figure 10 shows the trajectories followed for each robot and figure 11 shows the global map using CARMEN. Robot 1 explores across corridor 4 and Robot 2 explores corridor 3, each robot built its partial map and calculates its pose 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. 14).

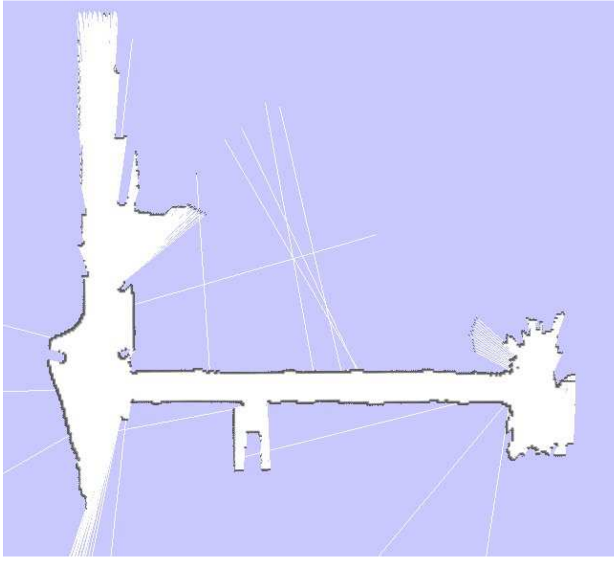

Fig. 13. Partial map built by robot 1.

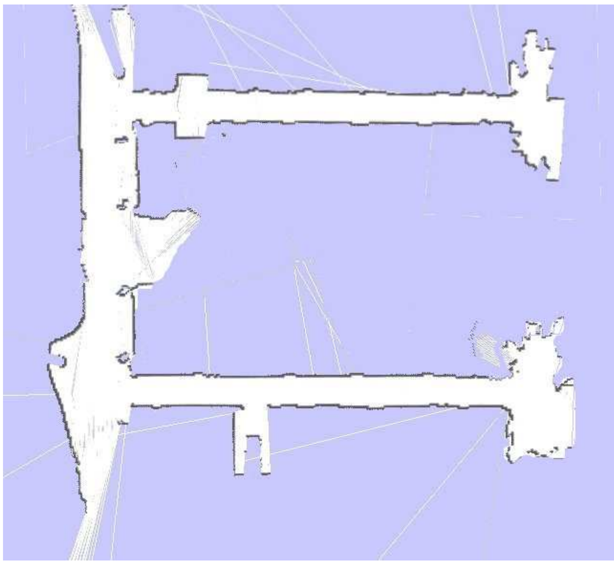

Fig. 14. Global map using scan-matching.

VII. CONCLUSION

We presented an initial work in mobile robot mapping. We have commented and show results using several algorithm for SLAM (scan-matching, grid-fastslam and grid-based slam with rao-blackwellized particle filter) for a single robot and multirobot map merging using scan-match technique. The results shows how is possible to use a team of robots to explore and navigate in unknown environments.

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.