Noname manuscript No.
(will be inserted by the editor)

Supervision and monitoring of logistic spaces by a cooperative robot team: methodologies, problems, and solutions

Received: date / Accepted: date

Abstract Mobile robots can be employed in the logistic field to efficiently perform common tasks, like building and updating maps of indoor and outdoor logistic spaces, locating specific goods on the map, tracing the product flow in the area, while preserving situational awareness and safety of the environment. This paper reports and discusses the main results of the MACP4Log (Mobile Autonomous and Cooperating robotic Platforms for supervision and monitoring of large LOGistic surfaces) research project, aimed at the study and development of a set of algorithms and services, enabling autonomous navigation of a team of mobile robots in large logistic spaces, and exploiting cooperation, through communication with a supervisor and among the robotic platforms. Although the main services required for the robots coincide with the most common issues of mobile robotics (i.e., localization, mapping, SLAM and exploration), the particular characteristics of the logistic spaces introduce specific problems (e.g., related to a high symmetry of the environment and/or to its variability), which must be properly taken into account. The paper discusses in detail such problems, summarizing the main results achieved both from the methodological and the experimental standpoint, and is completed by the description of the general functional architecture of the whole system, including navigation, logistic, and monitoring services.

Keywords Cooperative mobile robots · Logistic spaces · Mapping · Localization · Active SLAM and exploration

1 Introduction

The most recent years have witnessed a growing presence of robots and automated systems in the logistic field [1] - [6]. Interesting examples are given by the system for automated movement of shipping containers implemented in the Port of Brisbane (Australia), constituted by 18 autonomous straddle carriers (AutoStrad) [2], and by the *Kiva Mobile Fulfillment System* [63], which represents a more flexible, alternative solution for material handling than the classical Automated Storage and Retrieval Systems (ASRS) [20].

In indoor environments, the most common applications are related to pallet handling, and can make use of different kinds of Automated Guided Vehicles (AGV) [21]. The crucial point is constituted by the actual autonomy of such vehicles [22], and by the conditions in which they can effectively operate. In widespread commercial systems, independently of the navigation system used, predefined transport routes are generally followed and/or the presence of suitable landmarks is expected in the logistic area [64]. In both cases the environment is totally or at least partially structured, and preliminary operations must be carried out in the area, to create proper walkways, signs and lines on the floor (as path to be followed), and/or to locate *reflectors* (*targets*), strategically mounted on available landmarks (like racking and walls). Both solutions have some drawbacks. In the first case, epoxy type coating and sealers are generally used to create the lines: when their visibility becomes too poor (sometimes after only few months) and/or each time new lines must be added, parts of the warehouse could be out of action for some hours (or even up to a couple of days, depending on the used materials and to the dimensions of the involved area). Moreover, good illumination conditions are fundamental to avoid failures in the system. The possibility to use computer vision to let an AGV operate without a supporting structure in a warehouse has been also investigated [5]. In the second case, the use of targets, which must be made of high visibility reflective tape on ad hoc supports, can guarantee a proper localization of a vehicle only if at least three targets are detected at any time during travel, if the AGV works in full autonomy. Liu et al. [23] recently surveyed the most common automated guided methods. The presence of more AGVs in the area is allowed (path planning strategies and obstacle avoidance procedures prevent possible collisions), but they do not generally act as an autonomous coordinated team: their correct operativity is guaranteed through severe scheduling and routing procedures [24]. Only recently the possibility of using a swarm of small AGVs has been investigated [25] using a wireless network for their correct localization.

Significant improvements could be achieved on the one hand making the navigation system more robust with respect to changes in the routing grid [1], on the other hand enhancing its autonomy from the structural (or ad hoc created) characteristics of the environment. The use of a multi-robot cooperative system, composed of autonomous mobile robots, not requiring the presence of predefined landmarks or signed paths, seems to be the solution for more efficient applications in the logistic field. Multi-robot solutions give the possibility of performing complex or distributed tasks and increasing robustness thanks to the physical and architectural redundancy. Navigation, and the main functions it requires (like localization and mapping), can be performed effectively also in an initially unknown and potentially unstructured environment, thanks to the possibility to propagate the information acquired by each robot to all the team by means of a proper communication network. Moreover, the coordinated motion of all the team allows to explore and monitor the status of large areas in a shorter time. Some preliminary results have been already obtained by researchers in the logistics field, but only in limited applications or in simulation tests [4], [6].

The research project *MACP4Log - Mobile Autonomous and Cooperating robotic Platforms for supervision and monitoring of large LOGistic surfaces* [59] was aimed at the study and development of a set of algorithms and services, enabling autonomous navigation of a team of mobile robots in large logistic spaces, and exploiting cooperation, through communication with a supervisor and among the robotic platforms.

The main services to be performed by the robot team address the principal issues of potential logistic users and can be grouped into (i) proper logistic services and (ii) monitoring services. The first ones are based on the possibility of locating specific items on a predetermined map of the logistic space, such as containers, cars, etc., that can be marked by proper tags (e.g. license plates, bar codes, RFID) or unmarked, and should be distinguished by color, shape and/or other physical characteristics. The monitoring services instead are referred to general surveillance operations, such as the detection of possible intruders, and processing of heterogeneous data, like video stream from cameras, data about positioning, logs, alarms, etc.

In order to successfully accomplish such operational (logistic and monitoring) services, each mobile robot must be able to autonomously navigate, i.e., it must be able to: (i) build and update a map of the logistic space, (ii) self-locate within the given/constructed map, mainly by means of on-board sensors and cameras, (iii) travel to arbitrary places in the environment, avoiding static or moving obstacles, to perform its specific task, according to a general, efficient task allocation procedure among all the robot agents.

The overall goal was to develop a multi-robot system characterized by a high flexibility, not based on an expensive infrastructure, able to operate in an environment that can be initially not known and potentially unstructured, exploiting as much as possible the robot team coordination. Substantially flat indoor (or outdoor) spaces were considered, without using in any case GPS-based location sensors, while a wireless communication architecture was assumed available.

The particular characteristics of the logistic spaces and the aims of the project introduce specific problems that must be tackled in order to assure safe and reliable autonomous navigation:

- due to the high symmetry of the logistic areas (since the stored goods are often organized in regular grids), localization can become a difficult task without an absolute reference like GPS [26]: in absence of specific *ad hoc* landmarks, similar corridors between the goods can result undistinguishable, preventing the correct localization of each robot;
- since the environment in which the robot team has to operate can be initially not known and potentially unstructured, SLAM must be performed in a cooperative and efficient way, even if the initial position of the robots is not known or predefined, and the robots can communicate only within a limited range;
- since the dimensions of the logistic environment may be very large, during the SLAM phase it is fundamental to apply active strategies for trading-off between the contrasting tasks of exploring new parts of the unknown scenario and satisfying given con-

straints on the admissible uncertainty in the map estimation process, i.e., the so-called problem of *active SLAM and exploration* must be tackled;
- service robotic applications in logistic areas have to deal with intrinsically dynamic environments: since the goods to be tracked can be removed and substituted by other items many times during the day, the map must be periodically updated, including the detected variations, to continue to properly localize the robots without going back to a SLAM phase, as if the environment were completely different.

This paper reports and discusses the main achieved results of the research activity, both from the methodological and the experimental point of view.

The adopted robotic platforms for the experimental tests are Pioneer P3-DX mobile robots (the team is composed by three rovers, distinguishable by means of a bar code, as in Figure 1), equipped with proximity sensors, given by the embedded sonar ring and by a laser range finder SICK LMS200; a vision sensor (a pan-tilt or an omnivision camera, and/or a Kinect); broadband wireless transmitter and receiver, and an onboard PC. It is worth noticing that the robots P3-DX can be unsuitable for some industrial environments, where more robust platforms should be adopted; in this work they are used as test prototype to evaluate the effectiveness of the implemented algorithms and services.

Fig. 1 The robot team.

All the proposed solutions and approaches for the analyzed problems have been intensively tested by simulation and experimental tests. In particular:

- simulation tests have been performed in indoor environments, including the most critical characteristics of a logistic area (like a high symmetry) and/or in realistic scenarios, whose maps are well-known to the robotic community (for sake of repeatability and comparison);
- experimental tests have been carried out in indoor logistic-like environments, specifically reconstructed into rooms, labs and corridors of Politecnico di Torino or into larger spaces like a gym;
- indoor and outdoor tests have been performed in proper logistic scenarios (Figure 2), in the warehouse of *Prodit s.r.l.* in Santena (Torino, Italy) [65] and in the car deck of the *Ignazio Messina & C. S.p.A.* company in Genoa (Italy) [66], respectively.

Fig. 2 Experiments in a logistic warehouse.

The results of the simulation and experimental tests are reported and discussed in detail in previous papers [18], [19], [29]-[31], [36]-[38], [45]-[48]. Here, summaries of some of the main results can be found at the end of each section or subsection of the paper dealing with the specific topic (e.g., mapping, localization, etc.) considered in the test. Several photos and videos are available in the "Media" section of the project website [59]. Citations of specific videos of interest, relative to the different topics, will be given in the next sections.

The paper is organized as follows: Section 2 sketches the functional architecture of the whole system; Section 3 illustrates the approaches proposed to guarantee the main robots services required to accomplish various tasks, i.e., mapping, localization, path planning and task allocation; Section 4 describes the main methodologies proposed for the management of a logistic space. Section 5 draws some final conclusions.

2 System architecture

In the proposed functional architecture of the whole system (sketched in Figure 3), a Graphical User Interface (GUI) provides a user friendly way for a human operator to communicate with the robot team. The human operator can get through the GUI information about the status of the team of robots (e.g., the position of each robot, what type of task it is performing, etc.), and send to the team requests to accomplish one or more tasks, along with their associated priorities.

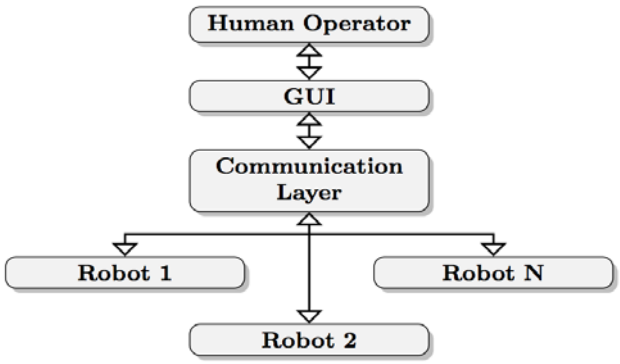


Fig. 3 The functional architecture of the system.

The GUI is bidirectionally connected with the communication layer, that manages and distributes messages among the robots, and allows the whole team to perform coordinated tasks. The blocks "ROBOT 1", "ROBOT 2" up to "ROBOT N" embody the functional architecture from the point of view of the single robot, and the content of each block is sketched in Figure 4.

At the top level of the functional architecture for each generic robot, the Robotic Agent block gathers the information coming from all the services provided by the robot, such as navigation and operational services, and broadcasts them towards the GUI.
The Navigation Services block guarantees the coordination and information sharing between the Localization, Map Building and Updating, and Path Planning and Task Allocation services. The Map Building and Updating Service provides active SLAM and exploration (mainly when building the first map of the environment) and map updating during the operativity of the logistic site. The Localization Service provides the correct localization of each robot thanks to a cooperative action with the other teammates, and through a WiFi communication infrastructure. High level tasks are allocated by the robotic agent sending the relative information to the Path Planning service, that builds local and global plans for the robot team.

The Operational Services include Logistic Services, such as physical tracking of goods and warehousing of their related information (e.g., position in the map, type of the good, etc.) as well as Monitoring services, such as detection of intruders, visual inspection of the status of the logistic site and video streaming of particular situations of interest.

Sensors onboard the robot and/or placed in the environment provide a data flow to Localization and Map Building and Updating Services, as well as to the operational services blocks.

The system described in this work has beed developed in C++ under Linux using the ARIA library [69]. For image capture and processing the OpenCV library [70] has been used. The robots are connected to a mobile ad-hoc network and messages are sent and received using multicast over TCP/IP. In order to optimize routing we used an implementation of the Optimized Link State Routing Protocol (OLSR) [28]. The whole system is able to run in near real-time on standard PCs, with the robots moving at speeds up to 1 m/s. The source code for the algorithms which are not included in non-disclosure agreements with partners will be soon released as ROS (Robot Operating System [67]) nodes.

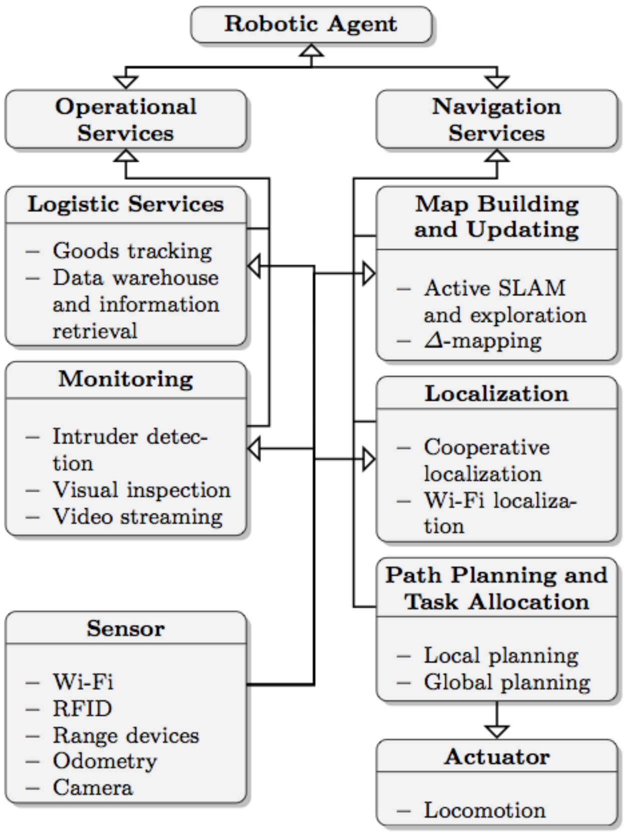


Fig. 4 The functional architecture of the single robot.

3 Navigation services supporting robot operation

After the team of robots is deployed in the logistic space, it is required to perform tasks in an unknown and potentially unstructured environment. It is possible to identify three main services which are necessary for robotic navigation: (i) to build and to mantain a representation of the surrounding environment (*mapping*); (ii) to estimate the robot position with respect to the environment map (*localization*); (iii) to plan a strategy to reach a predefined goal and to efficiently allocate the various sub-tasks among the members of the robot team (*path planning* and *task allocation*). Sections 3.1-3.3 discuss the aforementioned aspects, focusing on the solutions developed to tackle the particular characteristics of the logistic spaces, discussed in Section 1.

3.1 Map building: multi-robot SLAM and solutions for active SLAM and exploration

Before starting to perform any task, the robot team is required to build a map of the logistic space in which it operates. The importance of environment modeling is threefold: the map of the area is necessary to plan effective motion strategies; it can enhance situation awareness to the human operator who is going to monitor the logistic space via human machine interface (HMI); moreover it can be an intuitive tool for the operator to issue commands to the robots (reach target on the map, check the presence of a given good, detect intrusion in a specified area, etc.). The problem of map estimation without exact knowledge of robot position has been treated extensively in robotics, and the corresponding framework is usually referred to as *Simultaneous Localization and Mapping* (SLAM). If the logistic space is very large (as it often happens), in order to efficiently build the entire map in a short time it is fundamental (i) to take advantage of the cooperative action of the robot team and (ii) to make use of *active SLAM and exploration* approaches, i.e., planning robot motion in order to maximize the explored areas and at the same time minimize the uncertainty in SLAM. Exploration is clearly accomplished by visiting unknown places, whereas uncertainty reduction requires the robot to perform *loop closing* actions, i.e., to come back to already visited areas. The main results achieved for multi-robot SLAM and active SLAM and exploration are summarized in the two following subsections.

3.1.1 Rao-Blackwellized Particle Filters (RBPF) Multi Robot SLAM

In logistic applications it is fundamental that SLAM is performed in cooperative and efficient way, even if the area is quite large, the robots are initially deployed in random, unknown positions, and they can communicate only within a limited range.

In previous papers [30],[31] we proposed an application of Rao-Blackwellized Particle Filters (RBPF) for the purpose of cooperatively estimating SLAM posterior. There is a large literature in the field of multirobot SLAM, which ranges from the application of Extended Kalman Filter (EKF) [55] to Sparse Extended Information Filters [56], manifold representations [57], and RBPF [52]-[54], which is probably the most used approach to SLAM when the environment is described by means of a grid map. A deep discussion about the different SLAM approaches, referred in particular to the multi-robot case, is available in previous papers [30], [31]. Our contribution to the field consists in relaxing strict assumptions that characterize related work. We consider the realistic setting in which the relative initial positions of the robots are unknown and robots can communicate only within a limited communication range. In particular, they are required to exchange a small amount of information only when a rendezvous occurs and to measure their relative poses during the meeting. Before and after each rendezvous the robots of the team perform estimation using RBPF-SLAM [53], [54]. When a rendezvous occurs, the information is exchanged and efficiently fused, according to the following main steps of the proposed approach:

– *Data exchange*: the two robots, namely i and j, exchange the data acquired since the last meeting (or since the beginning if it is the first meeting between the two robots) to the rendezvous instant; each robot communicates only the odometric data and the corresponding laser scanner measurements;

– *Reference frame transformation*: from the information received by the teammate, and using relative pose measurements, each robot suitably roto-translates the data received in its own reference frame;

– *Estimation on virtual data*: once the data are roto-translated, they are used to estimate SLAM posterior as they were due to laser and odometric measurements acquired by the robot itself. RBPF estimate the posterior from received data, using suitable process models with the corresponding uncertainties.

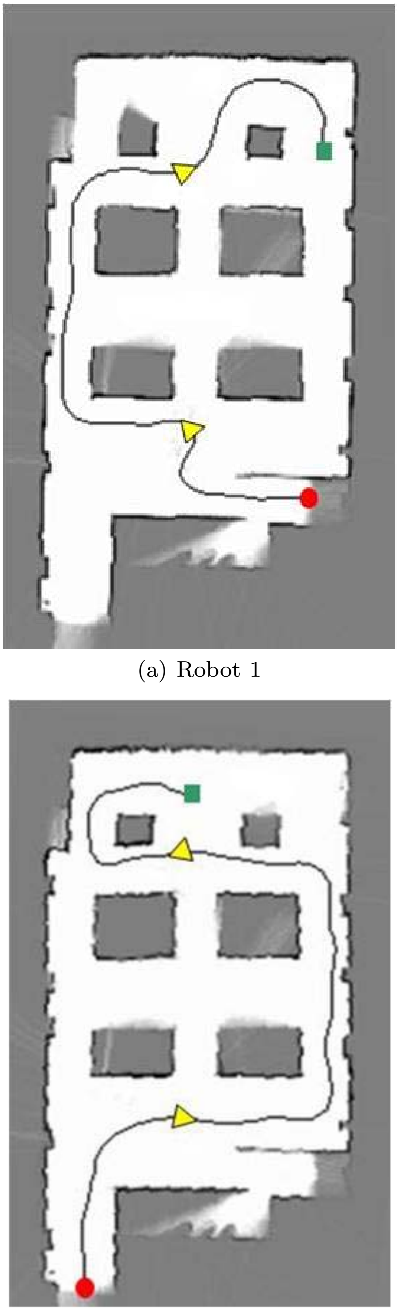
Finally, after filtering of received data is complete, the particles within the filter restart from their poses before the meeting, and continue the estimation process, according to grid-based RBPF-SLAM. Details can be found in a previous paper [31].

The described procedure can be easily generalized to an arbitrary number of meetings. After the first rendezvous each new meeting with a previously met robot corresponds to a loop closing event, adding constraints that are introduced in the filter through a resampling phase, which selects the trajectories that best describe all the acquired information. Moreover, including *vir-*

tual data from other robots, loop closure can occur also if the robot revisits places traveled by the met teammates. In the k-th rendezvous, each robot does not transfer the entire data set, but only the piece of information from the last meeting to the current time stamp, not only to limit the amount of exchanged data, but also to preserve filter consistency.

Several simulation and experimental tests have been performed, confirming the effectiveness of the approach. For example, Figure 5 shows the maps estimated by two robots, traveling in an environment of approximately 300 m^2, reproducing the structure of a logistic space. The trajectory followed by each robot is reported with a red circle denoting the starting point, a green square indicating the end of the path, and yellow triangles in correspondence of the rendezvous points, at which the robots were in line of sight with each other, so that they were able to measure their relative pose (and the corresponding uncertainty) using a camera associated with the laser range finder. Data were then exchanged, roto-translated in the own reference of each robot, and used as they were due to measurement acquired by the robot itself. In this way, even if none of the two robots physically traveled in the whole scenario, both of them succeeded in building a similar representation of the entire environment. As pointed out in a previous paper [31], the maps are not identical, due to the random nature of the RBPF. Moreover, the upper left part of the environment is different as it was only visited by robot 2 after the last rendezvous. However, in practice, rather than the similarity between the two maps, it is important to assure that the maps are faithful to the actual environment. In the same previous paper [31] we also report several simulated experiments (in which a ground truth map is available) and we show that the proposed approach produces maps that are faithful to the ground truth map, where map similarity is measured using the *acceptance index* proposed by Carpin [58].

A complete video of the experiment (titled "Multi-robot SLAM in indoor environments") is available [59].



(a) Robot 1

(b) Robot 2

Fig. 5 Maps estimated by the two Pioneer P3-DX during an experimental test in logistic space-like environment.

3.1.2 Active SLAM and Exploration with Particle Filters using Kullback-Leibler Divergence

During the phase in which SLAM is performed, it is necessary to quickly complete the coverage of the entire environment, while guaranteeing a good quality of the estimated map: a high uncertainty in the map would prevent the subsequent correct execution of robot tasks.

Active SLAM and exploration strategies can be applied to counterbalance the two opposite objectives of (i) visiting unknown places to extend the explored region and (ii) coming back to already visited areas (i.e., performing a *loop closing*) to improve the map quality. The active SLAM process requires metrics suitable for the evaluation of the uncertainty, so that proper bounds on the admissible uncertainty itself can be set. This metrics allow to define the so-called *information gain* of an action, which quantifies the advantage in performing a motion strategy (e.g., exploring new areas or reducing uncertainty). Although the evaluation of the *expected gain* of an action can be computationally demanding, effective and well founded approaches exist for EKF-based approaches [41]- [43]. The problem of active SLAM and exploration with particle filters, instead, is not fully understood and recent literature on the topic remarks several drawbacks of naive entropy-based metrics [44].

The main contributions to active SLAM and exploration with Rao-Blackwellized Particle Filters (RBPF) achieved from the research activity of MACP4Log project are detailed in previous papers [45]-[47]. First, an application of Kullback-Leibler Divergence (KLD) [49] has been proposed with the purpose of evaluating the particle-based SLAM posterior approximation [45]. This metric can be applied to quantify the advantage that the robot gains when reaching a target point. Active SLAM and exploration can be seen in fact as an optimization problem in which, given the information acquired until the current time (included in the SLAM posterior), the robot has to choose an exploration target (i.e., a goal point to be reached) so that the gain is maximized. The most common approaches give priority to the candidate targets that maximize the number of not yet visited cells, addressing in practice only the exploration problem and neglecting the risk of incorrect map estimation. The proposed metric has been instead applied to define the so-called *expected information from a policy*, so to evaluate the probability of the map estimation to be successful if a certain motion policy is followed [46].

Let $I(m_t) = I\left(p(m|x_{1:t}, d_{1:t})\right)$ be the current *map information*, computed from the probability distribution $p(m|x_{1:t}, d_{1:t})$, available to the robot at time t, where m is the map, $x_{1:t}$ is the robot trajectory, and $d_{1:t}$ includes the available measurement and odometry data until time t. Assume that a reasonable prediction of the amount of information, which can be acquired when applying the motion policy π to reach a given exploration target, can be computed. Denoting with $I(m_{t+T(\pi)}) = I\left(p(m|x_{1:t+T(\pi)}, d_{1:t+T(\pi)})\right)$ the predicted map information after the target is reached, the *expected information from policy* π is defined as:

$$E[I(\pi)] = p(\pi)[I(m_{t+T(\pi)}) - I(m_t)] + \\ +(1 - p(\pi))[-I(m_t)] \tag{1}$$

where $p(\pi) = p(\xi(p(x_{t:t+T(\pi)}, d_{t:t+T(\pi)}) < \bar{\xi}))$ is the probability of having an error smaller than $\bar{\xi}$ in the posterior approximation [45], ξ being the Kullback-Leibler Divergence between the estimated posterior and the true posterior.

Different entropy-based information metrics [50] can be adopted to define the map information $I(m_t)$; in our approach we chose to define it more simply as:

$$I(m_t) := N_t \tag{2}$$

where N_t is the number of visited cells in the map at time t. With this choice equation (1) simplifies to

$$E[I(\pi)] = p(\pi)N_{t+T(\pi)} - N_t \tag{3}$$

where $N_{t+T(\pi)}$ is the number of predicted observed cells, computed at time $t+T(\pi)$, $T(\pi)$ being the time required to execute the motion policy π.

A further simplification and normalization can be introduced, observing that: (*i*) the term N_t is common to all the targets, and can then be neglected when comparing the gain at different targets; (*ii*) the expected information can be normalized by the length of the trajectory corresponding to the motion policy π. The *expected information from a policy* is then finally defined as:

$$F_{\mathrm{EI}} \doteq \frac{1}{\ell} p(\pi) N_{t+T(\pi)} \tag{4}$$

where ℓ is the distance to be traveled.

The results of several, extensive tests are reported and deeply discussed in a previous paper [48]. Typical indoor environments, as well as benchmarking scenarios belonging to SLAM literature, have been considered to compare the performances of the proposed F_{EI} metric approach with state-of-the-art techniques, like *joint entropy* [50], [51], *expected map information* [44], and a *naive* metric, simply defined as:

$$G_N := N_{t+T(\pi)} - N_t \tag{5}$$

Here we summarize only the main results obtained for the autonomous exploration tests performed in simulation on three well-known benchmarking scenarios, i.e., the ACES building at the University of Texas (which covers an area of about 45 m by 40 m) in Figure 6(a); 2) the Intel Research Lab in Seattle (with size of 28 m by 28 m), in Figure 6(b); and 3) University of Freiburg 079 (FR079) building (in which the main corridor length is about 36 m) in Figure 6(c). The results of the tests have shown that:

– The average acceptance index [58] (defined with respect to the available ground true maps, considering 10 repetitions of each test) obtained with F_{EI} was comparable or greater than the values resulting from all the other metrics for all the scenarios. The achieved average values for F_{EI} were: 0.77 for FR079, 0.80 for Intel, and 0.63 for ACES.

- The number of failures in determining an acceptable map with F_{EI} was the lowest one in all the cases, and in particular no failure occurred for the FR079 building, while for the ACES building the proposed approach resulted to be the only one able to produce an acceptable map in 40% of the tests (i.e., 6 failures over 10 repetitions), whereas all the other techniques failed (9 or 10 failures over 10 repetitions). This scenario was in fact the most challenging one, since the presence of crossroads requires an effective exploration strategy able to recognize the possibility of closing a loop (it is the scenario more similar to a logistic-like environment), while the other two have structures which intrinsically force the robot to revisit known places.

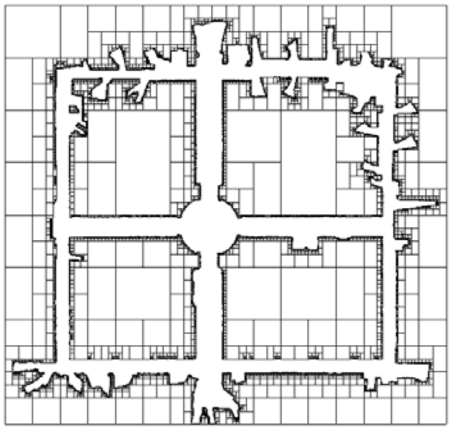
It is finally worth to notice that not only a poor exploration strategy may affect the map quality, but also the reverse implication holds: a poor map estimate may easily lead to exploration failure. An example of this situation is reported in Figure 7: the robot is not able to correctly close the loop, and in the estimated map it appears a new corridor that needs to be explored; in the real map, however, the robot is in a known area.

3.2 Robot localization and map updating

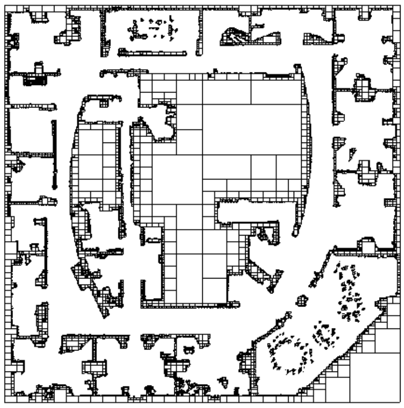
Once the map of the environment has been collaboratively created by the team, the correct localization of the robots with respect to such a map must be maintained, since it represents a prerequisite necessary to all the main tasks of the team of robots.

Potentially the multi-robot case gives some interesting advantages, since the accuracy of the robots pose estimates can be improved by mutual detections, even if wireless communication and data sharing problems must be considered. Monte Carlo Localizaton (MCL) methods approximate the probability density to be estimated using a finite set of samples [12]-[15], while other methods employ unknown but bounded error models for the sensor measurements [16], [17].

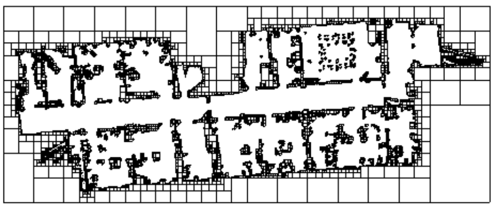
Unfortunately, without some external absolute information, a correct global self-localization cannot be performed in a short time by a single robot when the environment is highly symmetrical, like in a logistic area. For this reason multi-robot coordination can be used to improve the quality of self-localization estimates that single robots could achieve on the basis of their own sensors only. Subsections 3.2.1-3.2.2 sketch the proposed approach for our applications [36]-[38], benefitting on the one hand from the mutual position estimates coming from the other robots, and on the other hand from a WiFi infrastructure present in the environment [18].



(a) ACES building



(b) Intel Research Lab

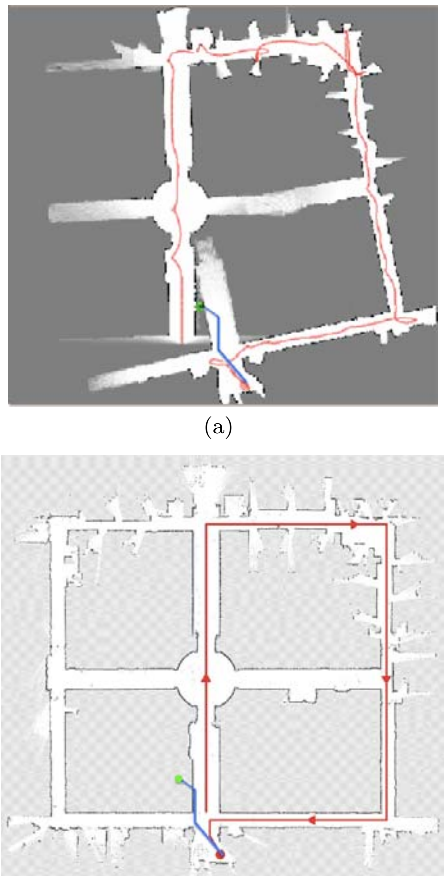


(c) Freiburg 079 building

Fig. 6 Maps of the benchmarking scenarios used for the comparative autonomous exploration tests.

Beside the high symmetry of the environment, there is a further specific characteristic of a logistic scenario that cannot be neglected in order to constantly guarantee the correct localization of the robots: the environment is intrinsically dynamic, since the goods stored in appropriate places can be removed and substituted by other items many times during the day. This implies the necessity of periodically updating the map, so to include the detected variations and continue to properly localize the robots without going back to a SLAM phase, as if the environment were completely different.

Subsection 3.2.3 illustrates the proposed Δ-*mapping* approach [19], [29], which allows the robots to detect



(a)

(b)

Fig. 7 (a) Estimated map with: estimated robot position (red dot), estimated robot trajectory (red line), and planned robot trajectory (blue line); (b) Ground truth map with: actual robot position (red dot), real trajectory (red line), and planned trajectory (blue line).

variations in the environment, generate a map containing only the persistent variations, propagate this map to the team and finally merge received information in a consistent way. Moreover team coordination is also exploited to assure the coverage of areas that have not been explored for long time, thus improving the knowledge on the present status of the map.

3.2.1 Multi-robot Localization in Highly Symmetrical Environments

The main contributions to the problem achieved during the MACP4Log research activity can be found in some previous papers [36]-[38]. The approach we proposed is based upon a particle filter cooperative Monte Carlo Localization (MCL) method and implements a three-stage procedure for the global localization and the successive position tracking of each robot of the team. The algorithm is able to exploit small asymmetries in the environment and spread this knowledge among team members, thus enhancing speed of convergence and robustness. Our approach extends the MCL approach proposed by Fox et al. [12] by introducing a three-state machine; the proposed solution does not use any absolute sensor data (e.g., GPS), that would be unavailable in indoor areas such as logistic warehouses. Online simulations and experimental tests, which investigate different situations with respect to the number of robots involved and their initial positions, show how the proposed solution can lead to the global localization of each robot, with a precision sufficient to be used as starting point for the subsequent robot tracking.

A preliminary solution [36] was proposed to solve the multi-robot localization problem using an algorithm based on two states only, but applied to a *completely* symmetric environment, with absolute heading measurements only occasionally available from a compass sensor.

We subsequently proposed the main approach, called *3SMCL* [37] and [38], which relies on distance measures only, under the assumption that the environment is not completely symmetric.

The first improvement with respect to the approach proposed by Fox et al. [12] is the use of a finite state machine, composed by three states: 1) GL = *global localization*, 2) UN = *undecided*, and 3) PT = *position tracking*, as shown in Figure 8.

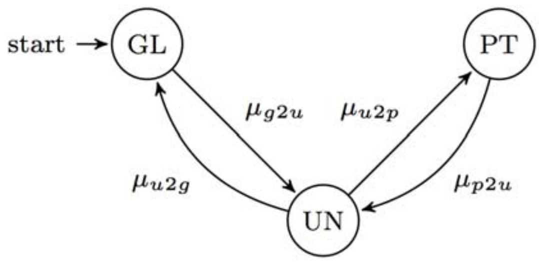


Fig. 8 The states and the thresholds.

The team members are able to detect each other by using visual markers (in our case one-dimensional barcodes were used) and send global position hypotheses to the detected robots over a communication network. Each robot then makes use of the received hypotheses according to its localization state. In the GL state the robot is performing global localization, so the received hypotheses are directly included in its particle filter; in the UN state the particle filter has already converged to few dominant hypotheses but the robot is unable to decide which one is the correct one. In the PT state the particle filter has only one dominant hypothesis and all the received hypotheses are in accordance with the one estimated by the robot. The passage between states is done using suitable *accordance* functions, as detailed hereafter.

The algorithm running onboard each robot is a typical Montecarlo localization approach [52]. The prediction phase is a classical one, and it computes a vector $p_i(t)$ containing the predicted pose (in terms of global coordinates $(x,\ y,\ \theta)$) for each particle.

Let $\mathcal{R} = \{r_i : i = 1, \ldots, N_R\}$ be the set of robots deployed in the area; with t we indicate the time variable that clocks the whole localization algorithm. Let k denote a time instant at which the position of the i-th robot is detected by a set of robots $R_i(k) \subseteq \mathcal{R}$, ($|R_i(k)|$ being its cardinality). The robots belonging to $R_i(k)$ send their measurements to the i-th robot, which collects them in the following matrix:

$$h_i(k) = \begin{bmatrix} \hat{x}_i^1(k) & \hat{y}_i^1(k) \\ \vdots & \vdots \\ \hat{x}_i^{|R_i(k)|}(k) & \hat{y}_i^{|R_i(k)|}(k) \end{bmatrix} \tag{6}$$

where $\hat{x}_i^j(k)$ and $\hat{y}_i^j(k)$, $j = 1, \ldots, |R_i(k)|$ are vectors containing the estimates of the position of the i-th robot expressed in Cartesian global coordinates. The position estimates contained in the j-th vector are generated starting from the position hypoteses of the j-th robot that has detected robot i. These position hypotheses are the result of a *Density-Tree* clustering [12] procedure applied to the set of particles.

The purpose of the update phase in our approach is twofold. It provides a vector $w_i(t)$ containing the importance factors for each particle, and it verifies whether the robot contains position estimates outside the map. If this is the case, their weight quickly decreases. At each time k, if $R_i(k)$ is empty, a classic *Kullback-Leibler Divergence* (*KLD*) resampling occurs [52]. If instead $R_i(k)$ is not empty, a modified version of the KLD resampling algorithm is applied (called *Mutual KLD* [38]). This modified resampling strategy allows to refine the probability distribution on the position of a robot using the estimates on its position given by other robots, and to determine the suitable number of particles needed to approximate the distribution according to the *KLD* distance. Consider a set of position hypotheses $h_i(k)$ of cardinality n_{hyp} received at time instant k; let N_{kld} be the number of particles used to approximate the belief on the pose of the robot without taking into account the position hypotheses $h_i(k)$. N_{max} denotes the maximum number of particles. If N_{kld} is greater than $N_{max} - N_{hyp}$, we reduce it to $N_{max} - N_{hyp}$, where N_{hyp} is the maximum number of particles that can be used in this case to approximate the probability distribution of the n_{hyp} position hypotheses. Then the suitable number of particles N^i_{mkld} needed to approximate the i-th hypothesis is calculated as

$$N^i_{mkld} = \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta} \tag{7}$$

where $\chi^2_{k-1,1-\delta}$ is a chi-square distribution with $1-k$ degrees of freedom. This value is the required number of particles to guarantee that with probability $1-\delta$ the *Kullback-Leibler* distance between the Maximum Likelihood Estimate (MLE) of the position hypothesis and the true distribution is less than ϵ.

At every instant robot i has a set of N_h hypotheses $\Phi_i(t) = \{\phi_i^j(t)\}$, $j = 1, \ldots, N_h$, and a best hypothesis $\phi^{best}(t)$. Each hypothesis is defined as $\phi_i^j(t) = \{p_i^j(t), \Sigma_i^j(t), W_i^j(t)\}$, where $p_i^j(t)$ is the predicted pose, $\Sigma_i^j(t)$ its covariance matrix, and $W_i^j(t)$ the associated weight, representing its confidence level.

The best hypothesis at time t is defined as $\phi_i^b(t) = \arg\max_{W_i^j}(\Phi_i(t)) = \{p_i^b(t), \Sigma_i^b(t), W_i^b(t)\}$. The center of mass of its own hypotheses is defined as

$$\bar{p}_i(t) = \sum_{j=1}^{N_h} \frac{p_i^j(t)}{N_h} \tag{8}$$

The mean distance among the hypotheses belonging to robot i is defined as

$$dist = \sum_{j=1}^{N_h} \frac{\sqrt{(\bar{x}_i(t) - x_i^j(t))^2 + (\bar{y}_i(t) - y_i^j(t))^2}}{N_h} \tag{9}$$

Switching rules among the three states are based on the following *accordance* function:

$$\mu_k = \sum_{q=k-n}^{k} \sum_{j=1}^{|R_i(q)|} \frac{\sqrt{(\hat{x}_i^j(q) - \hat{x}_i^b(t))^2 + (\hat{y}_i^j(q) - \hat{y}_i^b(t))^2}}{n|R_i(q)|} \tag{10}$$

where n is the length of the sliding window used to compute the average in (10). This quantity is an average on the distance between the hypotheses and the best position estimate of the i-th robot, averaged over the last n times the i-th robot has been detected. It must be noted that both (8) and (10) do not consider the weights of the hypotheses. This is due to the particular case of localization in highly symmetrical environments. In this situation, it can frequently happen that a position hypothesis with a high weight is not the correct one. Therefore, using weights in (8) and (10) may lead to wrong performance monitoring.

If the robot is in the UN state and it is deciding whether it can switch to PT, n is set equal to n_{u2p}. If the robot is in the UN state and it is deciding whether it has to switch back to GL, n is set equal to n_{u2g}. The inner summation in (10) averages the distances among the elements of $h_i(k)$ and the best position hypotheses of the i-th robot. The outer summation in (10) performs a moving average of length n on the results of the inner summation. Therefore μ_k measures the accordance between the actual belief on the position of the i-th robot and the average of the beliefs that the other robots have on its position at time k. When μ_k is lower than a certain threshold μ_{u2p} (empirically determined) the algorithm switches to PT. μ_k is computed also during the position tracking phase: if μ_k becomes greater than a given threshold μ_{p2u}, the algorithm switches again to UN. When the robot is in GL, it switches to UN if and only if the distance among the hypothesis defined in (9) is smaller than a certain threshold μ_{g2u}. The values of the various thresholds that are present in the algorithm are empirically determined, taking into account the environment characteristics, mainly its dimensions.

Since it is extremely difficult to formally prove the convergence of the particle filter to the true position of the robot, we provided extensive experimental validation about the reliability of the localization when the robots reach the PT state. It should be noted that the importance of knowing when all the robots are correctly localized is crucial from the application point of view, since only when all the team members are localized the application tasks assigned to the team can be carried out.

In order to validate the effectiveness of the proposed algorithm, we carried out a series of localization tests in both simulated and real environments to evaluate localization performances, scalability of the approach with respect to the number of robots in the team, and robustness to changes in the environment. The results are reported in a previous paper [38]; a short video, titled "Three-state Multirobot Collaborative Localization in Symmetrical Environments" is available [59]. Figure 9 reports the average localization error for a team of $N_R = 6$ robots over $n_l = 100$ localization runs in the same map used in the work by Fox et al. [12]. The map is shown in Figure 10 for reference. The robots were starting at random locations for each run and moved using a simple obstacle avoidance algorithm. We can see that the localization error decreases approximately linearly for all the robots, and the mean error among all the robots (dashed line in Figure 9) reaches a final value below 0.4 m.

As stated in a previous paper [38], the results are comparable with the approach proposed by Fox et al. [12]. It should be noted that in their work the simulated robots, as well as the recognition of the other robots, were ideal, while in our simulations we used a physical simulator, a simulated camera sensor and introduced appropriate sensor noise. We also show the results of $n_l = 5$ mutual localization experiments in a

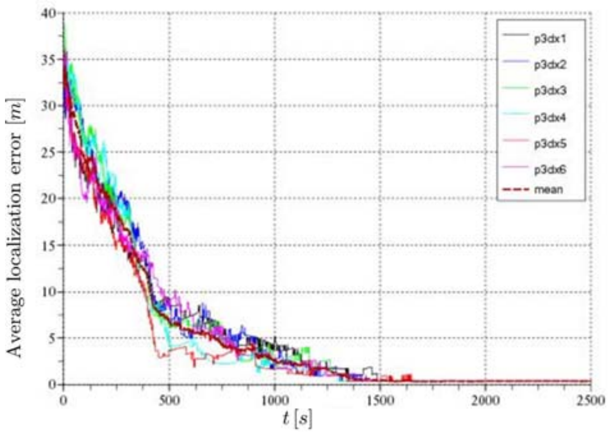


Fig. 9 Simulation test 1: average localization errors.

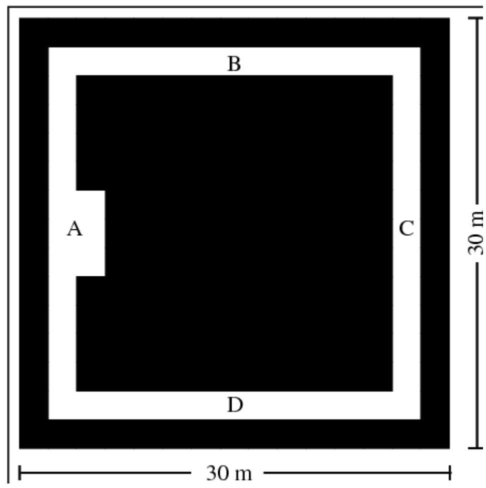


Fig. 10 Simulation scenario for test 1. White represents free areas, while black represents occupied areas.

real symmetric environment (see Figure 11) of dimensions 20×12 m using $N_R = 3$ robots randomly placed in the environment. Figure 12 shows the average localization errors. Each plot represents the average error among the 3 robots over one experiment. The ground truth to evaluate the localization error has been generated offline using our grid-based SLAM algorithm [31]. This algorithm also outputs laser-corrected odometry, which is accurate enough to be used as ground truth for the trajectory of the robot.

It can be noticed that the robots are able to spread information about the small asymmetry in the top-left corner and are able to correctly localize after 50 s.

3.2.2 Wi-Fi Based Robot Localization

In indoor locations GPS is generally not suitable to be used for global localization, since GPS signal is attenuated and scattered by roofs and walls; moreover, even

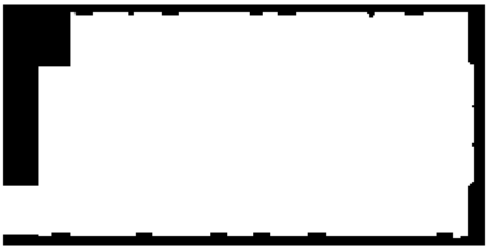


Fig. 11 Scenario for real experiments.

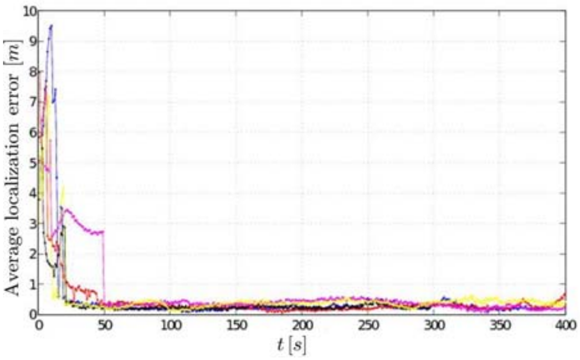


Fig. 12 Real experiments: average localization errors. Each plot represents the average localization error among the 3 robots over one experiment.

in outdoor areas, metallic canyons can negatively affect GPS transmission. On the other hand range-based localization can be subject to slow convergence and ambiguity due to symmetries, as discussed in Section 3.2.1.

In order to overcome these limitations, the use of Wi-Fi signal can be exploited. Wi-Fi networks are nowadays getting more and more widespread in many public places; we assume that a warehouse is already covered by several access points or such an infrastructure could be easily and inexpensively deployed. Moreover a Wi-Fi network is already employed for communication between rovers in the scope of the project.

In literature there are mainly two categories of localization systems using the Wi-Fi signal strength. In the first one an explicit mathematical model of the expected signal propagation is employed. These modeling methods are effective to determine a user or a rover location, but they rely on the knowledge of the Access Points (APs) position [39]. While this knowledge is easy to obtain in an industrial environment, we addressed a more general case in which the number and location of access points in the environment is unknown. Methods in the second category deal with Wi-Fi mapping strategies, and exploit received signal strength to infer position by using reverse functions based on fingerprinting and/or radio maps [40].

In a previous paper [18] we proposed a hybrid single-robot localization approach using both a Wi-Fi receiver and a traditional laser range finder. We first built radio maps modeling the signal strength of the APs as a function of robot position.

Then we investigated the problem of robot localization using two approaches. In the first approach we formulated the localization problem as a simple optimization problem, while in the second one integration with particle filters based localization was provided, too.

Given a set of APs, $A = \{A_k\}, k = 1, ..., |A|$, a set of radio-maps is built by interpolating the data coming from a sampling procedure. The set of radio-maps is denoted as $R^M = \{R_k^M, k = 1, ..., |R^M|\}, |R^M| = |A|$ where R_k^M represents a grid map indexed by l and m respectively for rows and columns, where $l = 1, ..., L$ and $m = 1, ..., M$.

Wi-Fi positioning can then be performed by comparing measured Received Signal Strength (RSS) values with the values stored in R^M and solving a minimization problem. At each time instant, given the RSS measured from the k-th AP at the robot location r_k, we define the matrix $\hat{R}_k^M = [\hat{r}_k^{l,m}]$ We define for each point of the map the Euclidean distance among the values in the $|R^M|$ radio maps and the corresponding measured values $\hat{R}_k^M$ as:

$$D_{l,m} = \sqrt{\sum_{k=1}^{|A|} (R_k^M - \hat{R}_k^M)^2} \tag{11}$$

According to the first proposed approach, the rover position is found as the minimum value of the surface $D_{l,m}$ using a simple gradient descent with negligible convergence times.

In the second approach, we proceeded to integrate the result of the Wi-Fi based localization technique with our particle filters based localization algorithm considering two levels of integration, a loose one and a tight one. In the first case WiFi-based localization provides a position estimate, with a fixed covariance error; in the second case the radio maps are used as additional inputs for the particle filter-based localization. The main motivation of these hybrid approaches is to exploit the advantages of both laser (high precision, but subject to ambiguity in symmetrical environments typical of logistic spaces) and Wi-Fi (less precise, but exempt from the previously mentioned ambiguity) positioning in order to achieve more efficiency in terms of precision, convergence, and reliability.

Experimental results show that both the proposed integrations improve the performances of the laser based localization, and in particular the tight integration with the particle filter achieves the best performances in localization [18]. We report in Figure 13 the results we

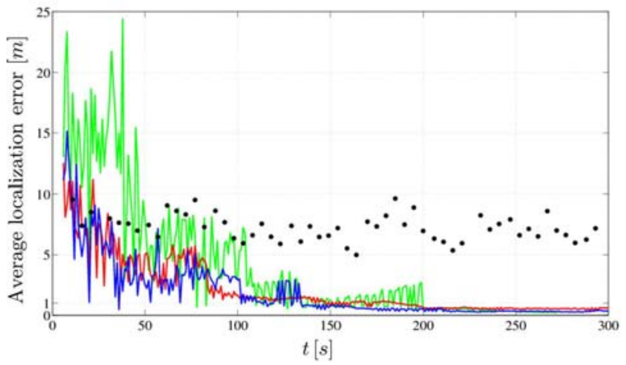


Fig. 13 Experimental test: average localization errors from the minimization approach (black dots), the laser-based approach (green line), the loose integration (red line), and the tight integration (blue line).

obtained in a large open environment. The area is rectangular (40 × 20 m) and obstacle-free. The black dots are the average localization error from the minimization approach, the green line is the result of the standard particle filter approach based on laser-scanner only, and finally the red line is the result of the loose integration, while the blue line is the result of the tight integration. The error values in the plot are averaged over $n_l = 10$ experiments. We can see that on average the error of the optimization approach is bounded, with values between 5 and 10 m. We also notice that on average both loose and tight integration approaches converge about 50 s sooner than the laser-based approach.

3.2.3 Multi-robot Map Updating in Dynamic Environments

Many real robotic applications require long-term mapping operativity in presence of *persistent* variations in the map, i.e., variations which alter the state of the environment not for very short periods, as in case of occasional people moving, but for long time intervals, as in the case of logistic scenarios, where the goods to be tracked can be removed and substituted by other items many times during the day. In some previous papers [19], [29] we proposed a methodology that allows the robots to detect variations in the environment, generate a map containing only the persistent variations, propagate this map to the team and finally merge received information in a consistent way. Moreover team coordination is also exploited to assure the coverage of areas that have not been explored for long time, thus improving the knowledge on the present status of the map. The map updating process is demonstrated to be computationally light, and can be performed in parallel with other tasks.

The team of mobile robots is supposed to be correctly localized with respect to the available environment map. This means that each robot is in the *position tracking* state [36], [37]. An occupancy grid map of the environment, previously created, is available to the robots. At discrete instants k the environment changes, and consequently the robots have to modify the map, to take into account the variation. We call this phase a *Δ-mapping* step.

We define the set of new maps collected by a robot up to time k as $\mathcal{M}(K) = \{M_k\},\ k = 0, \ldots, K$, where M_0 is the initial map, obtained by the SLAM procedure.

The goal of the algorithm is to provide an estimate $\hat{M}_k$ of the map at each time step k.

Furthermore, as the environment changes over time, it is desirable to keep track of the maximum possible number of variations, both for monitoring tasks (e.g., monitoring pallet movements in a storage area) and in order to minimize the number of *Δ-mapping* steps.

The architecture of the single-robot algorithm running onboard each robot is reported in Figure 14. More details regarding the different functional blocks can be found in a previous paper [19].

The *Δ-Awareness* block is able to detect persistent variations in the environment, using a technique called weighted recency averaging, which is normally applied in the problem of tracking non-stationary processes [52]. In our setting, the weighted recency averaging is employed to recognize changes in the environment, under the hypothesis that the robot is correctly localized and never kidnapped. When a persistent variation in the map is detected, the *Store scan* block starts to collect current laser scan readings l and current robot poses p and store them in matrices P and L

$$P = \begin{bmatrix} \hat{x}^1, \hat{y}^1, \hat{\theta}^1 \\ \vdots \\ \hat{x}^n, \hat{y}^n, \hat{\theta}^n \end{bmatrix} \tag{12}$$

$$L = \begin{bmatrix} l^1 \\ \vdots \\ l^n \end{bmatrix} \tag{13}$$

where the n-th entry is the last element stored. These matrices are used by the *Scan Alignment* block to create a local *Δ*-map containing the changes in the environment detected by the robot. Finally, the *Map Merge* block receives the updated map from the *Scan Alignment* block and merges it with the map that the robot is currently using for localization. The output of this merge process is a new map M_k.

The concept of *time-map* is introduced linking to each cell a value depending on the time passed after

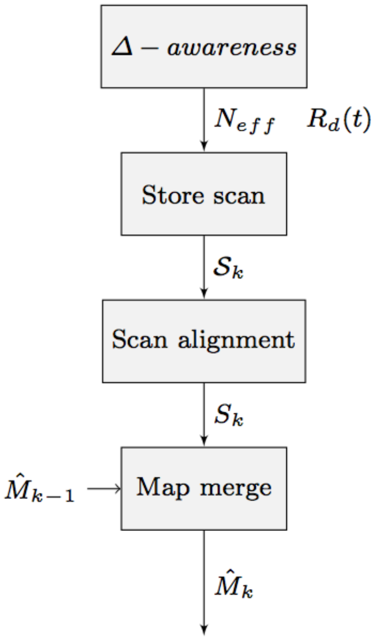


Fig. 14 Functional blocks of the *Δ-mapping* algorithm.

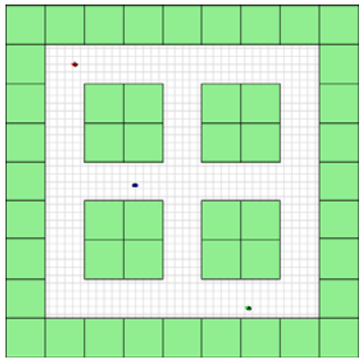


Fig. 15 The simulation environment for the *Δ-mapping* experiment.

the cell has been observed for the last time. High time-map values (close to 1) are assigned to cells belonging to recently mapped areas, while low values (close to 0) correspond to cells that have not been observed by the robot sensors for a long time.

The time map T_t is updated every time a laser scan is available to the robot; a ray tracing procedure is applied for each angle of the scan, assigning a maximum value equal to 1 to every cell crossed by a ray. At each time step all the values in T_t are updated according to

$$T_t(i,j) = T_{t-1}(i,j) \cdot \left(1 - \frac{\Delta t}{C_t}\right) \tag{14}$$

where Δt is the time elapsed from the last update of T_t, and C_t is a time constant which defines the forgetting speed.

When a new map M_k has been created by a robot, it is sent to the other team members via a communication network, including the current time map T_t.

When a robot receives a new map $\hat{M}'$ and a time-map T' from another robot, it merges the received time map with the previous map $\hat{M}_{k-1}$ and the local time-map T_t in order to produce $\hat{M}'_{k-1}$. T_t is also updated. For all the couples i, j the value of the cell $\hat{M}'_{k-1}(i,j)$ is set equal to the cell $\hat{M}'(i,j)$ if $T'(i,j) > T_t(i,j)$, otherwise it is set equal to $\hat{M}_{k-1}(i,j)$. The value of the cell $T_t(i,j)$ is set equal to $T'(i,j)$ if $T'(i,j) > T_t(i,j)$, otherwise it is not modified. Cells belonging to areas that have been recently mapped have high corresponding time-map values (close to 1), so recent changes in the map resulting from a local *Δ-mapping* process are not discarded.

A team coordination strategy that actively searches modifications in the map has been developed. Without any coordination strategy all the robots could follow the same path or leave some areas not visited for a long time. This problem can be treated in partial similarity with the problem of multi-robot exploration. In the exploration approaches the aim is to discover a map starting from a completely unknown environment. In the considered case, the initial map is known, as well as the robot pose, but since the environment is persistently changing (pallets are added and removed), the reliability of the initial map decreases over time on the basis of the number of changes in the environment. For this reason, areas that have not been recently visited may become completely unknown, as the reliability of the map in those areas is very low. Each robot keeps a time-map T_t updated at every step. Every time robot i receives a Δ-map M' from robot j, it receives also the time-map T'. The time-maps are then merged by keeping for each cell the one with the lowest value. Areas that need to be covered are the ones for which the corresponding value of the time-map is below a given threshold. For each robot, a set of points is extracted to feed the path planning algorithms from a topological map, which is constructed from the grid-map representing the areas to be visited.

The performances of the *Δ-mapping* process have been tested in long term operativity. The simulation scenario is sketched in Figure 15, where each block represents a container. A virtual fork-lift is present that adds and removes containers every two minutes. In this test the map updating process lasts for approximately 9.5 hours, for a total number of 328 variations. Figure 16(a) shows the localization error for a single run, while Figure 16(b) shows the acceptance index [58], which measures the accuracy of the map estimate, over 328 variations. The sudden increase of the localization error after approximately 6 hours is due to one of the robots losing its localization for a short period of time. This is due to the fact that the robot was traveling in an area with several variations. However as the robot receives an updated map it is able to recover itself. The acceptance index decreases to 0.97 after 9.5 hours, while in a similar experiment [19] the same error occurs after only 6 hours.

The video "Map updating in dynamic environments" [59], relative to a long term experiment, shows how the map is updated online, while the robot travels toward the targets that are subsequently assigned to it. The entire sequence of map variations over two hours is quickly reproduced in the last part of the video.

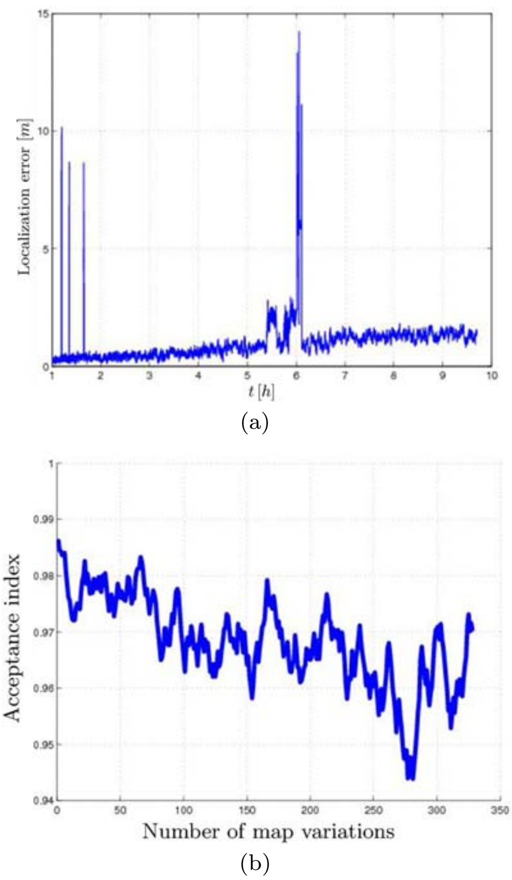


Fig. 16 Localization error (a) and acceptance index (b) for long-term *Δ-mapping* experiment

3.3 Path planning and task allocation

In a mobile robotic application, and in particular in a dynamic environment such as a logistic warehouse, one of the main prerequisites is the ability for the robots to travel avoiding static obstacles, like pallets or occupied areas, as well as moving obstacles like working machines, other robots, and people. At the same time the robots must be able to travel to arbitrary places in the environment, by following a path which is minimum with respect to travel distance, time of travel, risk of collision, maximization of the Wi-Fi coverage.

A two-level approach has been implemented for our application, due to its flexibility: a local path planner, which implements the VFH+ (Virtual Field Histograms) algorithm [33], is employed for obstacle avoidance, while a global path planner is built at a higher level using the wavefront algorithm [32].

The implementation of the VFH+ local path planner allows reaching goals relative to the robot local frame, using wheel odometry and a laser range finder while avoiding obstacles.

The global path planner is an implementation of the wavefront algorithm, which is able to find a sub-optimal path from a starting point to any reachable goal, under the assumption that a 2D grid-map representation of the environment is provided. The main advantages of the algorithm are that it is well-suited for grid-map representations of the environment and that it is usually faster than optimal approaches, such as A^*. Even if there is no guarantee of finding the optimal path, the algorithm surely finds a solution if it exists, and this solution is demonstrated to be reasonably near to the optimal one [32]. Once a suitable path from the robot position to the desired goal has been found, it is sub-sampled by discarding points which are reachable from previous ones with along a straight line by the robot (taken into account the radius of the robot and its kinematics). Such a procedure provides a set of waypoints, which are fed to the local path planner to obtain a smooth path.

This two-level path planning approach has been used throughout all the simulated and real experiments that have been presented in the previous sections.

Besides navigation and mapping the robot may be required to perform different tasks in the logistic space. Having a set of tasks, each task must be assigned to one of the robots in an efficient way in order to minimize some cost function. Two main constraints characterize the considered scenario: Wi-Fi based communication between robots is not always reliable (e.g., some robot can be temporarily out of range and unable to communicate with other team members), and task allocation strategies should be robust with respect to possible faults during task execution (e.g., a robot that is not responding due to hardware or software problems should not interfere with the other members of the team).

In order to deal with these issues we adopted a distributed market-based approach for our system. Our implementation is based on auctions, and it has been developed starting from the one proposed by Smith [35]. Every goal point is assigned to an auction using a multicast network channel; the robots reached by the auction compute and send back a bid. The auctioneer assigns the task to the robot with the best bid. The bid is computed according to the robot current position and its tasks queue. This approach does not guarantee the optimal solution, but is more robust to communication failures. Moreover in our approach the role of the auctioneer is always given to a different robot, thus avoiding the problem of single point of failure. The only constraint is that the internal clocks of the team members should be roughly synchronized, because the auction is valid only for a short period of time during which all the robots must answer with a bid. Bids that arrive once the auction is closed are discarded.

In our system we implemented three kinds of tasks: GOTO, PATROL, EXPLORE. While they all consist in reaching a particular place in the map, the behavior of the robot is slightly different for each task. For the GOTO task the robot has to reach a particular pose in the environment; for the PATROL task the robot has to reach a given pose and start patrolling the surrounding area. Finally, for the EXPLORE task the robot has to reach a given pose and then start wandering (random motion).

This task allocation service has been applied to all other services, in particular to the *Δ-mapping* service, and in all the experiments carried out throughout the project, proving to be fast and reliable in real scenarios.

4 Operational services

The management of a logistic space involves several tasks which can be carried out autonomously by the robotic team presented so far. Goods in the warehouse need be tracked, flow of products recorded and presence of people and intruders has to be monitored. In our implementation we consider two main service typologies that robots can provide, namely *logistic services* and *monitoring services*.

The formers are relevant for system management and goods tracking. We analyze a scenario in which goods (containers, pallets, or single products) are endowed with RFID (Radio Frequency IDentification) tags. The term RFID refers to a communication technology in which units, univocally distinguished through identifiers, can exchange information by means of radio frequency, within a certain range, also without line-of-sight between communicating devices. RFID technology is based on two main elements:

– Tag: it is a radio-frequency transponder, that incorporates an integrated circuit, connected with an antenna, with a simple control logic unit and limited storage capability (usually hundred of bytes or few Kbytes);

– Reader: it is a controller device that interrogates tags and retrieves information from them. It often provides measurements of Received Signal Strength Index (RSSI), which can be used to obtain a rough estimation of reader-tag distance.

In the last few years, RFID technology has received great attention for object identification and tracking, and it is currently used in several logistic applications (see for instance [7], [8], and the references therein).

Goods tagging provides a reliable and unambiguous methodology for robots to identify product flow in the environment [9]. Each robot is equipped with a RFID reader and is able to detect the presence and measure the distance from tags within a nominal range of $10 - 100$ m (active tags). Range measurements with RSSI are imprecise and affected by several parasite phenomena such as signal fading, reflection and multipath. On the other hand once the robots are well localized within the environment, the positions of tags can be efficiently estimated via maximum likelihood or other well known techniques (see our previous works on range localization [10], [11]). In Figure 17 it is possible to observe an example of RFID position estimation, in which each good is shown with a red dot superimposed on the map of the logistic space. In such a scenario the operator can monitor the position of the products, ask the system to indicate in which area a good is stored, and let the robot reach a given product. The robot team, while traveling, detects the tags and communicates to a control station all the information retrieved. Hence the team keeps trace of the product flow, which is recorded in a natural, inexpensive, and autonomous way.

The second service typology, namely the monitoring service, includes activities such as intruder detection, video streaming and access control. Although not strictly related to the logistic application, this service is crucial for enhancing situation awareness of the human operator who has to supervise the system. Access control can be easily performed when the employees are endowed with radio badges for accessing the logistic area, as it often occurs. Under this assumption, people presence and identification can be tackled with the same approach reported for goods tracking. Cameras provide the visual information needed by the operator and can be used for surveillance when the logistic space is in idle phases. It is worth noticing that the use of camera associated with RFID technology (badge or tags) allows to

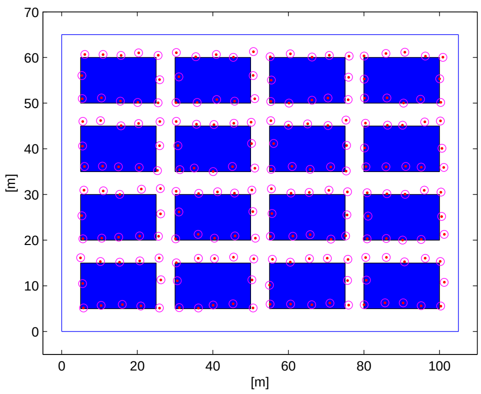


Fig. 17 Tag position estimation. We simulate the case in which each pallet (blue square) in the logistic scenario contains 12 items, each one tagged with an active RFID. During navigation, a mobile robot can detect tags and measure their distance using the on-board RFID reader. Then, using the approach described in [11], tags' positions are estimated from the distance measurements. Estimated tag positions are shown as magenta dots in the figure.

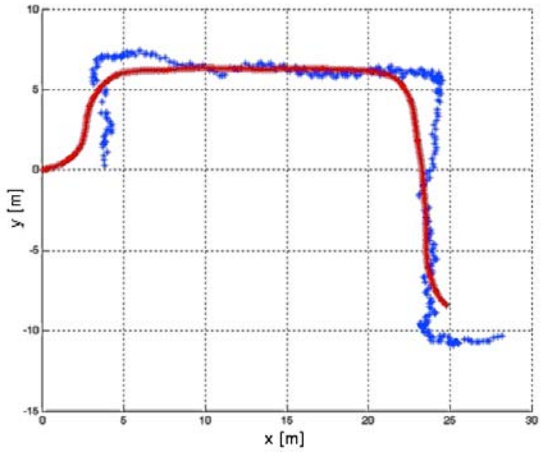


Fig. 18 User following with Kinect sensor. Red points represent the robot trajectory; blue points represent the position of the user over time relative to the robot position.

perform efficient and reliable intruder detection: moving objects which are not identified by a proper RFID device are classified as an intruder and the robot activates a streaming video towards the operator via HMI, to let him verify the nature of the unexpected presence. Recently, successful tests of people tracking and following have been carried out by using a Microsoft Kinect mounted on the mobile robot. The user is detected using Histogram of Oriented Gradients features [27] and recognized using linear Support Vector Machines. The distance of the user relative to the robot is then obtained directly from the Kinect sensor. In this way, the robot is able to track the position of the user in 3D.

Figure 18 reports the results of an experiment in which the robot is able to track and follow a user in real-time along a series of corridors; some videos are available, too [59].

5 Conclusions

The paper has reported and discussed the main results of the MACP4Log research project, aimed at the study and development of a set of algorithms and services, enabling autonomous navigation of a team of mobile robots in large logistic spaces, to cooperatively carry out operational services.

Since the project was not aimed for a specific application, it has been possible to consider various scenarios, analyzing and developing different strategies to achieve solutions suitable for a variety of possible problems and difficulties in logistic environments. At the same time, the lack of a particular, fully defined application to be performed has left some practical issues only partially figured out. The main ones are relative to the adopted robotic platforms, which are suitable quite exclusively for indoor environments (as revealed by the tests carried out in the car deck of the *Ignazio Messina & C. S.p.A.* company), not sufficiently robust for any industrial environment, and lacking of any manipulation ability, since no robotic arm has been mounted on them. At the same time it must be underlined that, even if the project was targeted to logistic environments, the developed algorithms and strategies can be easily adapted to other applications.

Current research is aimed at the implementation of the developed mapping, localization and exploration approaches into a ROS framework (Robot Operating System [67]) for monitoring of a data center by means of a robot team equipped with suitable sensors, like thermal cameras. ROS nodes will be soon publicly released.

Acknowledgements The project was funded by *Regione Piemonte* and coordinated by the Robotics Research Group from Politecnico di Torino [60]; the research activity was carried on in collaboration with *ERXA s.r.l.* [61], a SME specialized in software for industrial robotics applications, and the *Istituto Superiore Mario Boella (ISMB)* [62], a research institute active in several ICT fields.

The authors would like to thank all the people who participated and contributed to the successful results of the MACP4Log project, and in particular Fabrizio Abrate, Jingjing Du, Miguel Kaouk Ng, Vito Macchia, and Federico Tibaldi.

References

1. Barbera, H.M., Quinonero, J.P.C, Izquierdo, M.A.Z., Skarmeta, A.G. (2003) I-Fork: a flexible AGV system using topological and grid maps. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, vol. 2, pp. 2147 - 2152
2. Durrant-Whyte, H., Pagac, D., Rogers, B., Stevens, M., Nelmes, G. (2007) Field and service applications - An autonomous straddle carrier for movement of shipping containers - From Research to Operational Autonomous Systems. IEEE Robotics Automation Magazine 14(3): 14 - 23
3. Echelmeyer, W., Kirchheim, A., Wellbrock, E. (2008) Robotics-Logistics: Challenges for Automation of Logistic Processes. In: Proc. of the IEEE Int. Conf. on Automation and Logistics, pp. 2099 - 2103
4. Gao, J., Hu, X., Wu, C. (2006) Design and Simulation of Multi-Robot Logistic System. In: Proc. of the 2nd IEEE/ASME Int. Conf. on Mechatronic and Embedded Systems and Applications, pp. 1 - 6
5. Kelly, A., Nagy, B., Stager, D., Unnikrishnan, R. (2007) Field and service applications - An infrastructure-free automated guided vehicle based on computer vision - An Effort to Make an Industrial Robot Vehicle that Can Operate without Supporting Infrastructure. IEEE Robotics Automation Magazine 14(3): 24 - 34
6. Stiefelhagen, M. and Meijer, B.R. and van der Werff, K. and Tomiyama, T. (2004) Distributed autonomous agents, navigation and cooperation with minimum intelligence in a dynamic warehouse application. In: Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics, vol. 6, pp. 5573 - 5578
7. Lodewijks, G., Veeke, H.P.M., De La Cruz, A.M.L. (2006) Reliability of RFID in Logistic Systems. In: Proc. of the IEEE Int. Conf. on Service Operations and Logistics, and Informatics, pp. 971 - 976
8. Angeles, R. (2005) RFID techonologies: supply-chain applications and implementation issues. In: Information Systems Management, pp. 51-65
9. Niu B., Li, L., Ma, L., Xue, B. (2008) Design of Logistic Monitoring Management System Based on RFID Technology. In: Proc. of the Int. Symp. on Intelligent Information Technology Application Workshops, pp. 669 - 672
10. Calafiore, G., Carlone, L., Wei, M. (2010) A distributed Gauss-Newton approach for range-based localization of multi agent formations. In: Proc. of the IEEE Multi-Conf. on Systems and Control (MSC), pp. 1152 - 1157
11. Carlone, L., Bona, B. (2009) A comparative study on robust localization: fault tolerance and robustness test on probabilistic filters for range-based positioning. In: Proc. of IEEE Int. Conf. on Advanced Robotics, pp. 1 - 8
12. Fox, D., Burgard, W., Kruppa, H., Thrun, S. (2000) A Probabilistic Approach to Collaborative Multi-Robot Localization. Autonomous Robots 8(3): 325 - 344
13. Göring, D., Burkhard, H.-D. (2006) Multi Robot Object Tracking and Self Localization Using Visual Percept Relations. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 31 - 36
14. Peasgood, M., Clark, C., McPhee, J. (2005) Localization of Multiple Robots with Simple Sensors. In: Proc. of the IEEE Int. Conf. on Mechatronics and Automation, pp. 671 - 676
15. Rekleitis, I., Dudek, G., Milios, E. (2003) Probabilistic Cooperative Localization and Mapping in Practice. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1907 - 1912
16. Di Marco, M., Garulli, A., Giannitrapani, A., Vicino, A. (2003) Simultaneous Localization and Map Building for a Team of Cooperating Robots: A Set Membership Approach. IEEE Trans. on Robotics and Automation 19(2): 238 - 249
17. Taylor, C.J., Spletzer, J. (2007) A Bounded Uncertainty Approach to Cooperative Localization Using Relative Bearing Constraints. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2500 - 2506
18. Brevi, D., Fileppo, F., Scopigno, R., Abrate, F., Bona, B., Rosa, S., Tibaldi, F. (2009) Hybrid Localization Solutions for Logistic Applications. In: Proc. of the IEEE Int. Conf. on Technologies for Practical Robot Applications (TEPRA 2009), pp. 167 - 172
19. Abrate F., Bona B., Indri M., Rosa S., Tibaldi S. (2010) Map updating in dynamic environments. In: Proc. of the ISR/ROBOTIK 2010, Munich, Germany
20. Roodbergen K. J., Vis I.F.A. (2009) A survey of literature on automated storage and retrieval systems. European Journal of Operational Research 194: 343 - 362
21. Vis I.F.A. (2006) Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research 170: 677-709
22. Guizzo E. (2008) Three Engineers, Hundreds of Robots, one Warehouse. IEEE Specturm 7: 27-34
23. Liu J., Wang Q., He R. (2012) A survey of automated guided methods. In: Proc. of the IEEE Chinese Control and Decision Conference, pp. 3459 - 3462
24. Qiu L., Hsu W.-J., Huang S.-Y-, Wang H. (2002) Scheduling and routing algorithms for AGVs: a survey Int. Journal of Production Research 40(3): 745 -760
25. Lategahn J., Muller M., Rohrig C. (2012) Global localization of automated guided vehicles in wireless networks. In: Proc. of the 1st IEEE Int. Symp. on Wireless Systems, pp. 7 - 12
26. Kootstra G., de Boer B. (2009) Tackling the premature convergence problem in Monte-Carlo localization. Robotic Autonomous Systems 57: 1107-1118
27. Dalal N., Triggs B. (2005) Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition 1: 886-893
28. Jacquet P., Muhlethaler P.; Clausen T.; Laouiti A.; Qayyum A.; Viennot L. (2001) Optimized link state routing protocol for ad hoc networks. In: Proc. of the IEEE Multi Topic Conference
29. Abrate F., Bona B., Indri M., Rosa S., Tibaldi F. (2013) Multi-Robot Map Updating in Dynamic Environments. Distributed Autonomous Robotic Systems, Springer Tracts in Advanced Robotics 83: 147-160
30. Carlone L., Kaouk Ng M., Du J., Bona B., Indri M (2010) Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 243 - 249
31. Carlone L., Kaouk Ng M., Du J., Bona B., Indri M (2011) Simultaneous Localization and Mapping Using Rao-Blackwellized Particle Filters in Multi Robot Systems. Journal of Intelligent & Robotic System 63: 283 - 307
32. LaValle, S. M. (2006) Planning Algorithms. Cambridge University Press
33. Ulrich, I.; Borenstein, J. (1998) VFH+: reliable obstacle avoidance for fast mobile robots. In: Proc. of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 1572-1577
34. Kuhn, H. W. (1955) The Hungarian method for the assignment problem. Naval Research Logistics Quarterly vol. 2, pp. 83 - 97

35. Smith, R. G. (1981) The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers C-29(12): 1104 - 1113
36. Abrate F., Bona B., Indri M., Rosa S., Tibaldi S. (2008) Switching Multirobot Collaborative Localization in Symmetrical Environments. In: Proc. of the IROS 2008, 2nd Workshop on Planning, Perception and Navigation for intelligent Vehicles, pp. 94-99
37. Abrate F., Bona B., Indri M., Rosa S., Tibaldi S. (2009) Three-State Multirobot Collaborative Localization in Symmetrical Environments. In: Proc. of the 9th Conference on Autonomous Robot Systems and Competitions, pp. 1-6
38. Abrate F., Bona B., Indri M., Rosa S., Tibaldi S. (2013) Multirobot Localization in Highly Symmetrical Environments Journal of Intelligent & Robotic Systems 71(3): 403-421
39. A. Smailagic and D. Kogan (2002) Location sensing and privacy in a context-aware computing environment. Wireless Communications 9: 10-17
40. C. Rohrig and F. Kunemund (2007) Estimation of position and orientation of mobile systems in a wireless lan. In: Proc. of the 46th IEEE Conference on Decision and Control, pp. 4932 - 4937
41. Bourgault, F., Makarenko, A.A., Williams, S.B., Grocholsky, B., Durrant-Whyte, H.F. (2002) Information Based Adaptive Robotic Exploration. In: Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, pp. 540 - 545
42. Martinez-Cantin, R., De Freitas, N., Doucet, A., Castellanos J.A. (2007) Active policy learning for robot planning and exploration under uncertainty. In: Proc. of Robotics: Science and Systems, Atlanta, USA
43. Sim, R., Roy, N. (2005) Global A-Optimal Robot Exploration in SLAM. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 661 - 666
44. Blanco, J.L., Fernandez-Madrigal, J.A., Gonzalez, J. (2008) A novel measure of uncertainty for mobile robot SLAM with Rao-Blackwellized particle filters. Int. J. Robot. Research 27(1): 73 - 89
45. Carlone, L., Kaouk Ng, M., Du, J., Bona, B., Indri, M. (2009) Reverse KLD-Sampling for Measuring Uncertainty in Rao-Blackwellized Particle Filters SLAM. In: Proc. of the Workshop on Performance Evaluation and Benchmarking for Next Intelligent Robots and Systems, IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, USA
46. Carlone, L., Du, J., Kaouk Ng, M., Bona, B., Indri, M. (2010) An Application of Kullback-Leibler Divergence to Active SLAM and Exploration with Particle Filters In: Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, pp. 287 - 293
47. Du, J., Carlone, L., Kaouk Ng, M., Bona, B., Indri, M. (2011) A Comparative Study on Active SLAM and Autonomous Exploration with Particle Filters. In: Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, pp. 916 - 923
48. Carlone, L., Du, J., Kaouk Ng, M., Bona, B., Indri, M. (2013) Active SLAM and Exploration with Particle Filters using Kullback-Leibler Divergence", Journal of Intelligent & Robotic Systems (online October 2013) doi: 10.1007/s10846-013-9981-9
49. Kullback S., Leibler A. (1951) On Information and Sufficiency. Annals of Mathematical statistics. 22: 79-86
50. Stachniss C., Grisetti G., Burgard W. (2005) Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In: Proc. of Robotics: Science and Systems
51. Roy N., Burgard W., Fox D., Thrun S. (1999) Coastal Navigation: Mobile Robot Navigation with Uncertainty in Dynamic Environments. In: Proc. of Int. Conf. on Robotics and Automation, pp. 35-40
52. Thrun, S., Burgard,W., Fox,D. (2005) Probabilistic Robotics. MIT Press
53. Grisetti, G., Tipaldi, G.D., Stachniss, C., Burgard, W., Nardi, D. (2007) Fast and accurate SLAM with Rao-Blackwellized particle filters. Journal of Robotics and Autonomous Systems. 55: 30 - 38
54. Doucet, A., de Freitas, J., Murphy, K., Russel, S. (2000) Rao-Blackwellized particle filtering for dynamic bayesian networks. In: Proc. of the Conference on Uncertainty in Artificial Intelligence, pp. 176-183
55. Zhou, X.S., Roumeliotis, S.I. (2006) Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1785-1792
56. Thrun S., Liu, Y. (2005) Multi-robot SLAM with sparse extended information filters. Robotics Research, pp. 254-266
57. Howard, A., Sukhatme, G.S., Mataric, M.J. (2006) Multi-robot simultaneous localization and mapping using manifold representations. Proceedings of the IEEE, 94: 1360-1369
58. Carpin, S. (2008) Fast and accurate map merging for multi-robot systems. Autonomous Robots 25(3): 305 - 316
59. MACP4Log project: `http://www.polito.it/macp4log` (2013)
60. Robotics Research Group (RRG), Politecnico di Torino: `http://www.polito.it/LabRob` (2013)
61. ERXA: `http://www.erxa.it` (2013)
62. Istituto Superiore Mario Boella: `http://www.ismb.it` (2013)
63. Kiva Systems: `http://www.kivasystems.com` (2013)
64. Logistics Handling: `http://www.logisticshandling.com` (2013)
65. Prodit: `http://www.prodit.com`
66. Ignazio Messina & C. S.p.A.: `http://www.messinaline.it`
67. Robot Operating System (ROS): `http://wiki.ros.org` (2013)
68. The Player Project `http://playerstage.sourceforge.net/` (2013)
69. Aria library `http://www.mobilerobots.com/software/aria.aspx` (2013)
70. OpenCV library `http://opencv.org/` (2013)