Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 53 (2005) 282–311
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory

Rui Rocha[a,b], Jorge Dias[a,*], Adriano Carvalho[b]

[a] *Institute of Systems and Robotics, Faculty of Sciences and Technology, University of Coimbra-Pole II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal*
[b] *Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*

Received 24 September 2004; received in revised form 19 August 2005; accepted 6 September 2005
Available online 2 November 2005

Abstract

Building cooperatively 3-D maps of unknown environments is one of the application fields of multi-robot systems. This article addresses that problem through a probabilistic approach based on information theory. A distributed cooperative architecture model is formulated whereby robots exhibit cooperation through efficient information sharing. A probabilistic model of a 3-D map and a statistical sensor model are used to update the map upon range measurements, with an explicit representation of uncertainty through the definition of the map's entropy. Each robot is able to build a 3-D map upon measurements from its own range sensor and is committed to cooperate with other robots by sharing useful measurements. An entropy-based measure of information utility is used to define a cooperation strategy for sharing useful information, without overwhelming communication resources with redundant or unnecessary information. Each robot reduces the map's uncertainty by exploring maximum information viewpoints, by using its current map to drive its sensor to frontier regions having maximum entropy gradient. The proposed framework is validated through experiments with mobile robots equipped with stereo-vision sensors.

Keywords: Multi-robot systems; Cooperation; 3-D mapping; Stereo-vision sensors; Entropy; Information utility; Frontier-based exploration

1. Introduction

Multi-robot systems (MRS) have been widely investigated for the last decade [1–4]. These systems employ teams of cooperative robots to carry out missions that are either inherently distributed in time, space or functionality, and cannot be achieved by a single robot, or wherein a multi-robot solution is more efficient, cost effective, reliable and robust than a single robot solution. Cooperation has, in general, three main potential advantages: (i) *efficiency*—taking advantage from the spatial distribution of sensors makes possible to reduce the map's uncertainty more quickly than if a single robot is used; (ii) *reliability and robustness*—with re-

* Corresponding author.
E-mail addresses: rprocha@isr.uc.pt (R. Rocha), jorge@isr.uc.pt (J. Dias), asc@fe.up.pt (A. Carvalho).

0921-8890/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2005.09.008

dundancy in robots capabilities, the failure of any particular robot does not necessarily compromise the overall mission success; (iii) *specialization*—robots with different sensory or motion skills may have complementary and specialized features that overcome their individual limitations and increase the team's total utility. Building a 3-D map of an unknown environment is one of the application fields of MRS.

This article addresses two main issues: (i) developing a probabilistic model for vision-based 3-D mapping and frontier-based exploration using information theory and (ii) sharing information efficiently through communication in a team of cooperative mobile robots, driven by information utility maximization.

1.1. Robotic mapping

Robotic mapping addresses the problem of acquiring spatial models of physical environments with mobile robots, which might be used to safely navigate within the environment and perform other useful tasks (e.g. surveillance). Some examples of sensors used for building maps are cameras, range finders using sonars, laser or infra-red rays, radars, tactile sensors, etc. As sensors have always limited range, are subject to occlusions and yield measurements with noise, mobile robots have to navigate through the environment and build the map iteratively. Some key challenges arise from the nature of measurement noise (sensor modeling problem), high dimensionality of the entities being mapped (representation problem), the correspondence or registration problem (registering measurements on a common coordinate space), dynamically changing environments and defining an efficient survey strategy to build the map (exploration problem) [5].

Robots can be used for building fastidious maps of indoor environments [6,7], but they are particularly useful on mapping missions of hazardous environments for human beings, such as underground mines [8–10], where updated maps are required to prevent future accidents related with inundations or collapses, but where humans access is too risky or even impossible due to difficult access routes; or nuclear facilities [11], where monitoring the state of the sarcophagus interior is required by maintenance procedures, but where humans exposure to radiation must be avoided.

1.1.1. Grid-based maps

Grid-based maps [12,13], also known as occupancy grids or certainty grids, are widely used to intuitively represent distributed spatial information, such as occupancy or, closely related, traversability. They discretise the workspace being mapped in cells with a given resolution. For each cell, it is maintained a probabilistic belief about its state (e.g. free or occupied). In Ref. [12], Moravec and Elfes developed occupancy grids as a space representation model. In their seminal work, they built 2-D occupancy grids by using a robot with sonars. In Ref. [6], they extended the occupancy grid technique for environment mapping of 3-D grids, using stereo-vision as primary sensor. Borenstein and Koren developed the vector field histogram [14], which is a popular obstacle avoidance method based on 2-D occupancy grids. Grocholsky et al. propose in Ref. [15] the integration of a decentralized architecture – Decentralized Data Fusion – with occupancy grids, as a means to combine observations from multiple robots with communication capabilities. In Ref. [16], it is described a blimp project where 3-D space is represented through digital elevation maps, which are 2-D grids associating height with each cell. In Ref. [7], the notion of occupancy grid was refined to avoid the binary representation of the cell's occupancy and to model it as a continuous value between 0 and 1. They used 2-D coverage maps to perform indoor exploration tasks with a robot equipped with sonars.

In this article, we propose a grid-based probabilistic model of a 3-D map, which stores for each cell (voxel) a coverage belief. Concerning the map's representation model, our main contribution is a more compact representation of this belief than using histograms [7], and a straightforward and efficient Bayesian update procedure. We also develop a method to easily update the map upon new data yielded by range sensors.

1.1.2. Registration, localization and SLAM

Robot's autonomous localization is tightly related with mapping, because accurate mapping depends on localization, which in turn relies on tracking the robot's position to distinguishable landmarks identified in the current map, if a global localization scheme is not available. During the mapping process, the robot has to register measurements obtained from different locations, which requires its ability to localize itself accurately in the map. Moreover, in a multi-robot solution, estimat-

ing accurately the robots' relative position is required to register measurements from different robots on a common coordinate space. There is some recent work on 3-D mapping with a single robot, focusing mainly on the registration problem [11,9].

The problem of building a map and simultaneously tracking the robot's position on that map is known as Simultaneous Localization and Mapping (SLAM). Extensive research has been devoted to SLAM for the past few years and important progress has been achieved [17,10,16,18–22]. It is an important issue, because it provides an integrated solution of localization and mapping for applications where a global localization system is not available and the robot is subject to accumulation of pose errors during mapping. Most of the proposed solutions are based on the implementation of an extended Kalman filter, which correlates localization estimates relative to different landmarks. Thrun et al. [10] approach mine mapping as a SLAM problem but, due to cyclic structure of mines, it yields difficult correspondence problems. To solve this problem, they use an iterative closest point algorithm, generating 3-D maps by applying scan matching to 3-D measurements after a 2-D occupancy grid map of the mine is obtained. In Ref. [18], the variant FastSLAM is presented, which combines a particle filter for sampling robot paths and an extended Kalman filter for representing the map. The particle filter implements a robust Monte Carlo localization algorithm [23]. This approach is more robust to data association problems than algorithms based on maximum likelihood data association.

There are also some efforts to develop multi-robot localization algorithms [24–27] and SLAM extensions to multi-robot systems [28–30]. Within these methods, when a robot determines the location of another robot relative to its own, both robots can refine their internal beliefs based on the other robot's estimate and improve localization accuracy. Fox et al. introduced a probabilistic approach based on Markov localization, which has been validated through real experiments showing a drastic improvement in localization speed and accuracy, when compared to single robot localization [24]. Roumeliotis and co-workers addressed the determination of upper bounds on the position uncertainty accumulation for a group of robots, by using an extended Kalman filter [25,26]. Martinelli et al. extended this approach, by considering the most general relative observation between two robots [27]. Sujan et al. proposed a SLAM-based architecture [31] to a cliff surface exploration mission with a robot team. Each robot repositioned its sensors using an information-theoretic approach so as to fill uncertain regions of the environment map, based on maximizing the expected new obtained information.

In this article, we assume that robots are externally localized through a global localization system, so the work here presented does not fall in the heading of registration, localization and SLAM [24,28,25–27,29,30].

1.1.3. Exploration and active sensing

When a robot or a team of robots explore an unknown environment and build a map, the objective is to acquire as much new information as possible with every sensing cycle, so that the time needed to completely explore it is minimized.

Bourgault et al. [32] used occupancy grids to address the single robot exploration problem, as a balance of alternative motion actions from the point of view of information gain (in terms of entropy), localization quality (using SLAM) and navigation cost. Although they include information gain in their strategy, their formulation is computationally heavy and they are only able to use it off-line, for a limited number of proposed destinations. Yamauchi proposed frontier-based exploration [33] whereby robots are driven towards boundaries between open space and unexplored regions. He also proposed a decentralized scheme whereby robots shared local 2-D occupancy grids, which were fused with their own local maps in order to obtain a global grid. Each robot explores the environment by selecting the closest frontier cell in its neighborhood. Burgard et al. developed a technique for coordinating a team of robots while they are exploring their environment to build a 2-D occupancy grid [34]. Their approach uses the frontier-cell concept proposed in Ref. [33] and considers a balance between travel cost and utility of unexplored regions so that robots simultaneously explores different regions. The utility of a region is reduced when a robot selects a target viewpoint whose visibility range covers it. They do not define an architecture for the team and it is not clear how robots should interact and what to communicate to accomplish the proposed coordination. In their seminal work reported in Ref. [35], they used entropy minimization to actively localize a robot by minimizing the expected future uncertainty.

In Ref. [36], the problem of merging local maps from different robots, with unknown start locations, is addressed. Robots that can communicate with each other are arranged in exploration clusters. The robots within each cluster share a common map and coordinate their exploration using an algorithm similar to the one proposed in Ref. [34]. Before two robots merge their local maps, they actively verify their relative locations, through the implementation of a particle filter and a rendezvous strategy. The solution has been applied within the Centibots project [37], which deploys 100 robots in unexplored areas to build a map, search for valuable objects and protect the environment from intruders. A similar project is presented in Ref. [38], reporting experiments with a team of 80 heterogeneous robots. In Ref. [39], a 2-D grid-based version of FastSLAM [18] algorithm is developed, which generates trajectories to actively close loops during SLAM and takes into account the uncertainty about the pose of the robot during the exploration. Whenever this uncertainty becomes too large, the robot re-visits portions of the previously explored area. When the localization uncertainty is low and no loop can be closed, a frontier-based exploration strategy [33] is used.

Our approach to exploration and active sensing is closely related with frontier-based exploration [33,34], with two important improvements. Firstly, we explicitly define a distributed architecture model for the robots, which restricts the communication among robots to the minimum necessary to share useful sensory data among robots and to coordinate the exploration. Secondly, we use entropy to explicitly represent uncertainty in the grid-based probabilistic map, as a means to define a formal information-theoretic background to reason about the mapping and exploration process.

1.2. Communication in cooperative multi-robot systems (MRS)

Most of the work in MRS has been devoted to the definition of different architectures [40,41,33,42,43,15,37], mostly behavior-based [44,45], that rule the interaction between the behaviors of individual robots. Communication is a central issue of MRS because it determines the possible modes of interaction among robots, as well as the ability of robots to build successfully a world model, which serves as a basis to reason and act coherently towards a global system goal. Communication may appear in three different forms of interaction [1]: (i) *via environment*, using the environment itself as the communication medium (stigmergy); (ii) *via sensing*, when an agent knowingly uses its sensing capabilities to observe and perceive the actions of its teammates; and (iii) *via communication*, using a communication channel to explicitly exchange messages among the agents, thus compensating perception limitations.

Arkin [46] demonstrated that sometimes cooperation between robotic agents was possible even in the absence of communication, though it is a weak form of cooperation, which may me very inefficient. Matarić [47] showed that the ability to distinguish other robots from the rest of other objects provides sufficient power to overcome interference. Balch and Arkin [48] made simulation studies of three typical multi-agent tasks, using the three basic communication types referred above, and found that: communication improves performance significantly in tasks with little implicit communication; and that more complex communication strategies (goal-oriented) offer little benefit over basic communication (state). Within CEBOT framework, Fukuda and Sekiyama [49] studied methods that sought to reduce communication requirements, by increasing the awareness level of individual cells. Parker [50] investigated the impact of awareness on a MRS and concluded that it improves performance, regardless of team size. Tambe presented STEAM [51], a general model of teamwork, which includes a heuristic that attempts to follow the most cost-effective method of attaining mutual belief in joint intentions, by managing a trade-off between communication and team incoherence costs. Stone and Veloso [52] proposed a method for inter-agent communication, which assumes that agents alternate between periods of limited and unlimited communication.

Although previous work on communication structures for MRS has led to some useful conclusions and design guidelines, there is no a principled formalism that can be systematically used to share efficiently sensory data based on information utility assessment, in order to support the efficient use of communication in MRS. As communication is always limited, either in resources applied to perceive the world or in bandwidth of a communication channel, using efficiently those resources is crucial to scale up coop-

erative architectures for teams of many robots, without limiting them to simple reactive and loosely cooperative systems, with very limited or no awareness. Current architectures extensively use explicit communication, not taking care [40,41,33,42,43], giving low emphasis [37], or using no principled heuristics to avoid the communication of redundant information. The work of Grocholsky et al. [15] is an exception to this trend, because they use entropy to define theoretic information measures for predicting the expected information outcome associated with control actions. Although it seems to be a rigorous method to model the information flow within a team of robots, it is not clear how it can be used to share efficiently sensory data within mapping missions, and it is mainly focused on coordination. We propose an information-theoretic measure of information utility which is used by robots to cooperate through sharing sensory data in 3-D mapping missions, without overwhelming communication resources with redundant or unnecessary information.

1.3. Notation

This sub-section states some notation that is used throughout the article, in order to improve its readability. Hereafter, vectors are written with bold lowercase letters; tuples and sets are both written with uppercase letters, but sets are written in calligraphy. Random variables (RV) are written with uppercase letters. The entropy of discrete RV X is $H(X)$ and $I(X;Y)$ denotes the mutual information between two RV X and Y. The joint entropy of a set of discrete RV $\mathcal{X} = \{X_1, \ldots, X_n\}$ is $H(\mathcal{X})$. The differential entropy of a continuous RV Y is $h(Y)$.

The fleet of n robots is the set $\mathcal{F} = \{1, \ldots, n\}$. The index $k \in \mathbb{N}_0$ is used for batches of measurements. The set of time instants when measurements are obtained is $\mathcal{T} = \{t_k : t_k \in \mathbb{R}, k \in \mathbb{N}_0\}$, with $t_{k-1} \leq t_k, \ \forall_{k\in\mathbb{N}}$. The kth batch of measurements is obtained at time instant $t = t_k \in \mathcal{T}$, being $t = t_0 \leq t_k, \ \forall_{k\in\mathbb{N}}$ the initial time instant, which is associated to index $k = 0$. The robot's pose is $Y = (\mathbf{x}, \mathbf{a})$, which includes the sensor's position $\mathbf{x} \in \mathbb{R}^3$ and its attitude $\mathbf{a}(t) \in \mathbb{R}^3$ (three Euler angles), whose angles are assumed to be positive in the counterclockwise direction. The robot's pose when the kth batch of measurements is obtained is $Y_k = (\mathbf{x}_k, \mathbf{a}_k)$ and the selected viewpoint (navigation target) for the robot is Y^s. Every coordinates refer to a global coordinates frame $\{W\}$.

The kth batch of measurements is $M_k = (\mathbf{x}_k, \mathcal{V}_k)$, being $\mathcal{V}_k$ a set of m_k applied vectors (measurements), in the point $\mathbf{x}_k$. The sequence of batches of measurements up to the kth batch of measurements is the set $\mathcal{M}_k = \{M_i : i \in \mathbb{N}, i \leq k\}$, being $\mathcal{M}_0 = \emptyset$ the initial empty sequence. The kth batch of measurements sent to other robots is $S_k = (\mathbf{x}_k, \mathcal{U}_k), \mathcal{U}_k \subseteq \mathcal{V}_k$, being $\mathcal{U}_k$ a set s_k of useful measurements. The kth batch of measurements received from other robots is $R_k = (\mathbf{x}'_k, \mathcal{U}'_k)$, being $\mathcal{U}'_k$ a set of u_k useful measurements from the sensor of other robot, whose pose is $\mathbf{x}'_k$.

The 3-D workspace is divided into equal sized voxels with edge $\epsilon, \epsilon \in \mathbb{R}$. The set of all voxels yielded by such division is the 3-D grid $\mathcal{Y}$. The index l is used for denoting individual voxels. The function $v : \mathbb{R}^3 \to \mathcal{Y}$ determines what grid's voxel a given 3-D point belongs to. The function $\mathbf{w} : \mathcal{Y} \to \mathbb{R}^3$ computes the center coordinates of a voxel $l \in \mathcal{Y}$.[1] The set of voxels traversed by a vector $\vec{\mathbf{u}}$ when applied in point $\mathbf{a}$ is denoted as $\mathcal{Z}(\vec{\mathbf{u}}, \mathbf{a})$.

The coverage of a voxel $l \in \mathcal{Y}$ is represented through the continuous RV C_l, taking values $c_l \in [0, 1]$. Given a batch M_k, the set of influenced voxels by a measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$ is denoted as $\mathcal{Z}_{k,i} \subset \mathcal{Y}$ and the measurement's information utility is denoted as $I_{k,i}$. The information utility of the kth batch of measurements is denoted as I_k. An individual measurement influencing the coverage estimate of a voxel $l \in \mathcal{Y}$ is the tuple $D^l_j = (d_j, d^l_j)$, being $d_j \in \mathbb{R}$ the distance between the sensor and the detected obstacle and $d^l_j \in \mathbb{R}$ the distance between the sensor and the voxel's center. The set of measurements influencing the coverage estimate of a voxel $l \in \mathcal{Y}$, after k batches of measurements, is $\mathcal{D}^l_k = \{D^l_j : j \in \mathbb{N}, j \leq n_k(l)\}$, having cardinality $n_k(l) \in \mathbb{N}_0$. Before the first batch, i.e. for $k = 0$, the set of influencing measurements is $\mathcal{D}^l_0 = \emptyset$. The coverage probability density function of a voxel $l \in \mathcal{Y}$, after k batches of measurements, is $p(c_l|\mathcal{M}_k) = p(c_l|\mathcal{D}^l_k), 0 \leq c_l \leq 1$. The voxel's entropy is denoted as $H(l) = H(C_l)$ and its gradient as $\vec{\nabla} H(l)$.

The 3-D probabilistic coverage map after k batches of measurements is the set of coverage random variables $\mathcal{C} = \{C_l : l \in \mathcal{Y}\}$ described statistically by the

[1] The voxel's center is a point equidistant to all voxel's faces, i.e. its geometric center.

set of pdf's $\mathcal{P}(\mathcal{C}|\mathcal{M}_k) = \{p(c_l|\mathcal{M}_k) : l \in \mathcal{Y}\}$, being $\mathcal{P}(\mathcal{C}|\mathcal{M}_0) = \{p(c_l|\mathcal{M}_0) = p(c_l|\mathcal{D}_0^l) : l \in \mathcal{Y}\}$ the initial map. The map's entropy after k batches of measurements is denoted as $H(\mathcal{C}|\mathcal{M}_k)$ or simply as $H(t_k)$. The mission execution time is $t_{k_{\max}}$, being associated to the last batch of measurements, i.e. the $k_{\max}$th batch of measurements. Given the entropy threshold H_{th}, it is the first batch of measurements for which $H(t_k) \leq H_{\text{th}}$. The associated total number of processed measurements is m_{T}, the total number of received measurements form other robots is r_{T} and the traveled distance by the robot during the mission is d_{T}. The mission execution time as a function of the number of robots n is $t_{k_{\max}}(n)$.

1.4. Organization of the article

This article is organized as follows. Section 2 presents the proposed distributed architecture model for 3-D mapping, giving an overview of the framework presented in the following sections. Section 3 presents the theoretical background concerning entropy and mutual information. Section 4 presents a probabilistic model of a 3-D map, which encompasses: updating the map upon new sensory information based on a Gaussian sensor model of a stereo-vision sensor, using entropy to assess the map's uncertainty, and using the map's entropy field to perform a frontier-based exploration of the environment. At the end of this section, after briefly describing an experimental setup comprised of mobile robots equipped with stereo-vision range sensors, some examples of volumetric maps are shown. Section 5 introduces a mutual information-based measure of information utility. Robots use this measure to cooperate through sharing information, by selecting and communicating the most useful information to their peers. After briefly describing the multi-robot software architecture, Section 5 ends with the presentation of results that demonstrate the benefit of cooperation through sharing sensory data. The article ends with conclusions and future research guidelines.

2. Architecture for 3-D mapping with a multi-robot system

Figs. 1 and 2 depict complementary views of our architecture model for 3-D mapping. Although individual robots belonging to the multi-robot system might be heterogeneous in terms of sensory skills and mobility, all of them follow the same architecture model when performing the 3-D mapping mission. That is why both figures refer to an individual robot; nevertheless, the interaction with other robots is represented through the communication block and its associated data flow.

Fig. 1 shows the different parts of the process and how they interact. The robot's platform is assumed to have a sensor, a localization module and an actuator. The sensor provides new sets of vectors $\mathcal{V}_{k+1}$ where obstacles are detected from the current sensor's pose $Y(t)$. The localization module gives the sensor's pose $Y(t)$, including position and attitude.[2] The actuator changes the sensor's pose (robot's pose) accordingly with a new selected exploration viewpoint Y^s. New data from the robot's sensor is associated with its current pose, given by the localization module, to form a new batch of measurements $M_{k+1} = (\mathbf{x}_{k+1}, \mathcal{V}_{k+1})$. Then, index k is incremented and the new batch of measurements becomes the current batch M_k. The memory of measurements is updated as $\mathcal{M}_k = \mathcal{M}_{k-1} \cup M_k$. The previous map $\mathcal{P}(\mathcal{C}|\mathcal{M}_{k-1})$ is updated upon the new batch of measurements M_k, which yields the current map $\mathcal{P}(\mathcal{C}|\mathcal{M}_k)$. The current map is used to choose a new target pose Y^s which is the reference input to the robot's actuator. As part of map updating, it is built a batch of measurements $S_k = (\mathbf{x}_k, \mathcal{U}_k)$ having the most useful data from sensor $\mathcal{U}_k \subseteq \mathcal{V}_k$. Those selected measurements are shared with other robots through the communication module. This module can also provide the robot with batches of measurements $R_k = (\mathbf{x}'_k, \mathcal{U}'_k)$ given by other robots and the map is updated accordingly. Cooperation among robots arises because of this altruistic commitment to share useful measurements.

Fig. 2 depicts a flowchart showing the sequence of the aforementioned robot's operations and interactions. At the beginning of the mission an initial map is given to the robot. Then it gets a new batch of measurements, updates the map and shares useful measurements with other robots. Then it might receive measurements from other robots and, in that case, the map is updated accordingly. Given the new map, a new viewpoint for the

[2] In this article, the localization problem is not addressed. It is assumed that each robot is able to localize itself in a global coordinates frame.

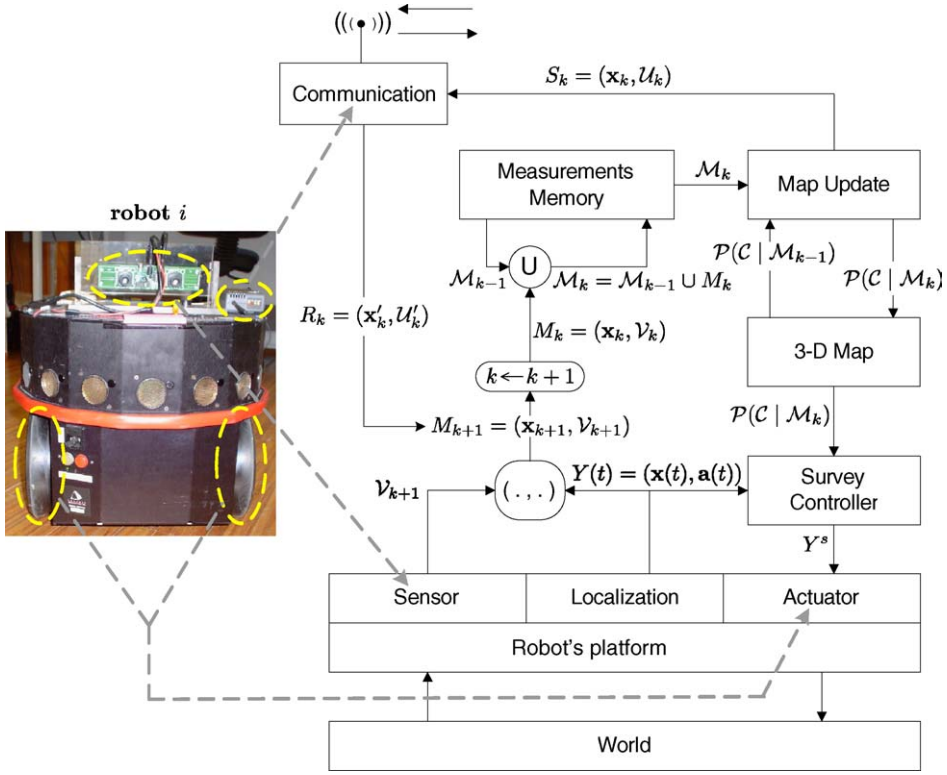

Fig. 1. Block diagram showing the relation between different parts of the process and the resources of a given robot i of the fleet $\mathcal{F}$.

sensor is chosen and the robot starts moving itself. During navigation, the robot continues updating the map whenever new data is received from other robots. When the robot reaches the new target pose, the process repeats itself with a new batch of measurements provided by the sensor from its new pose.

3. Entropy and mutual information

Entropy is a general measure for the uncertainty of a belief [53]. When applied to a discrete random variable, it evaluates to its shortest description, being as high as the variable's uncertainty [54]. Being X a discrete RV over a discrete sample space $\mathcal{S}$ with probability distribution $p(x) = P(X = x)$, entropy is defined as the expected value of $\log \frac{1}{p(X)}$:

$$H(X) = -\sum_{x \in \mathcal{S}} p(x) \log p(x) = E\left[\log \frac{1}{p(X)}\right]. \quad (1)$$

Notice that $H(X) \geq 0$, being assumed the continuity convention $0 \log 0 = 0$. The logarithm's base determines the information unit whereby entropy is measured. Hereafter, we use the base 2 for the logarithm and, in this case, entropy is measured in bits.

Given two discrete RV X and Y, the entropy definition can be extended to compute the joint entropy $H(X, Y)$ and the conditional entropy $H(X|Y)$ or $H(Y|X)$ [54]. For instance, the entropy $H(X|Y)$ is the entropy of X if Y is given. The *joint entropy's chain rule* theorem states

$$H(X, Y) = H(X) + H(Y|X), \quad (2)$$

$$H(X, Y) = H(Y) + H(X|Y), \quad (3)$$

$$H(X) - H(X|Y) = H(Y) - H(Y|X), \quad (4)$$

which means that joint entropy is the entropy of one variable plus the conditional entropy of the other. Given that X and Y are statistically independent RV if $p(x, y) = p(x)p(y)$, the following inequalities can be

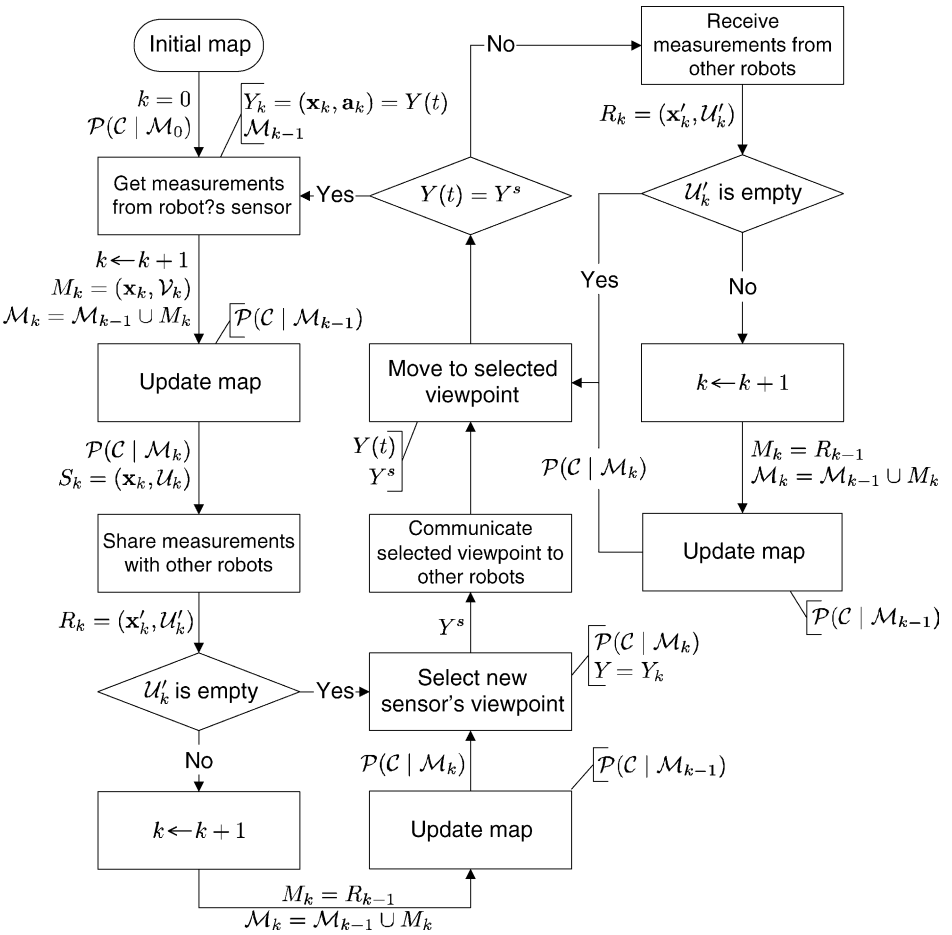

Fig. 2. Flowchart showing the data flow during the mission of a given robot i of the fleet $\mathcal{F}$.

proved:

$$H(X, Y) \leq H(X) + H(Y), \tag{5}$$

$$H(X|Y) \leq H(X), \tag{6}$$

$$H(Y|X) \leq H(Y). \tag{7}$$

Equalities occur when X and Y are independent RV.

Mutual information provides a measure of the reduction of a RV's uncertainty due to the knowledge of another [54] and it can be defined as

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{8}$$

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \tag{9}$$

Eq. (8) suggests that mutual information may be viewed as a measure of the statistical dependence between two random variables. The definitions provided by Eq. (9) states that mutual information is the information of a variable minus its information if the other is given. Note that $I(X; Y) = I(Y; X)$ and $I(X; Y) \geq 0$, where the equality occurs if X and Y are statistically independent RV. Since $I(X; X) = H(X) - H(X|X) = H(X)$, entropy is sometimes referred to as *self-information*. The *conditional mutual information* of two RV X and Y given another RV Z is defined as

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z), \tag{10}$$

which is a generalization of Eq. (8) to conditional distributions.

3.1. Differential entropy

Entropy's classical definition applies only to discrete RV because it was developed by Shannon as a measure of information for computer networks. However, its definition might be generalized for continuous RV, being denoted as *differential entropy* [54]. Being $f(x)$ the probability density function of a given continuous RV X with a continuous domain $\mathcal{S}$, it is defined as

$$h(X) = -\int_S f(x) \log f(x)\, \mathrm{d}x. \tag{11}$$

As probability density functions may evaluate to values greater that one, differential entropy cannot be taken as an absolute measure of information or uncertainty because it can be negative. Most of the properties and theorems of the entropy definition for discrete RV are also valid for differential entropy. However, while the latter tends to $-\infty$ when a RV has no uncertainty/information, the former evaluates to zero. The discrete definition is thus more convenient because it is always non-negative.

3.2. Sets of discrete random variables

The joint entropy chain rule theorem given by Eq. (2) can be extended to a set of more than two RV [54]. The *joint entropy* of a set of discrete RV $\mathcal{X} = \{X_1, \ldots, X_n\}$ with joint pdf $p(\mathcal{X}) = p(X_1, \ldots, X_n)$ is

$$H(\mathcal{X}) = H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_1, \ldots, X_{i-1}). \tag{12}$$

Using Eq. (6), it can be proved the inequality

$$H(\mathcal{X}) \leq H(X_1) + H(X_2) + \cdots + H(X_n) = \sum_{i=1}^{n} H(X_i), \tag{13}$$

wherein the equality occurs if all RV in $\mathcal{X}$ are independent.

4. Probabilistic volumetric maps

An important resource for robotic mapping and exploration is obviously a map representing the robots' knowledge about the environment. As sensors have limited range, are subject to occlusions and yield noisy measurements, robots have to navigate through the environment and build the map iteratively, in order to reduce the map's uncertainty. In this section, we propose a *grid-based probabilistic* representation of 3-D maps which enables to model explicitly uncertainty. Although the proposed framework might be used to model any phenomena spatially distributed, as we have validated it through experiments with robots equipped with stereo-vision range sensors providing distance measurements, hereafter we will denote a map as a coverage map, which is a 3-D representation of the environment occupancy with obstacles.

4.1. Definition of a volumetric model

One of the most popular space representation models are *occupancy grids*, which are discretised random fields wherein the probability of occupancy of each independent cell is maintained [12,13]. They have been extensively used in robotics mainly due to their simplicity and suitability for decision-theoretic approaches. Some recent examples of their application are [34,32,55]. The definition of probabilistic map that we use was first introduced in Ref. [7], wherein the notion of *occupancy grid* was refined in order to avoid a strictly binary representation of each cell's occupancy (free or occupied), through the notions of *coverage* and *coverage map*. The *coverage* of a cell is the portion of the cell that is covered by obstacles (a value between 0 and 1). A *coverage map* stores for each cell of a given grid a probabilistic belief about its coverage. Our innovation on the use of this space representation is the way we represent and update the voxel's coverage belief [56] and using it on 3-D mapping with cooperative multi-robot systems [57].

Our volumetric model [56] assumes that we define a 3-D discrete grid $\mathcal{Y}$, which divides the robotic team workspace into equally sized voxels (cubes) with edge $\epsilon \in \mathbb{R}$ and volume ϵ^3. Fig. 3 shows a geometric representation of our model. Any edge of any voxel is assumed to be aligned with one of the axes of a global coordinates frame $\{W\}$. The portion of the volume of voxel $l \in \mathcal{Y}$ which is covered (occupied) by obstacles is modeled through the continuous random variable C_l, taking values c_l in the interval $0 \leq c_l \leq 1$, and having $p(c_l)$ as its probability density function. The objective

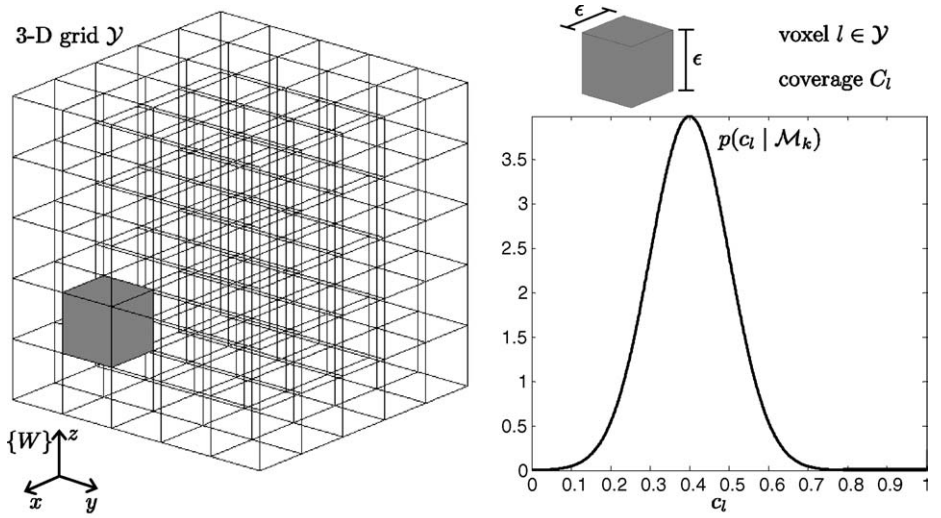

Fig. 3. Volumetric discrete grid: (a) the grid divides the workspace into equally sized voxels, whose edges are aligned with one of the axes of the world coordinates frame $\{W\}$ and (b) the coverage C_l of each voxel $l \in \mathcal{Y}$ with edge ϵ, given the sequence $\mathcal{M}_k$ of k batches of measurements, is modeled through a probability density function $p(c_l|\mathcal{M}_k)$ (the example is a normal pdf $N(\mu_l = 0.4, \sigma_l = 0.1)$).

of building a map is to obtain for each voxel $l \in \mathcal{Y}$ an estimate as accurate as possible about its coverage C_l. The advantage of modeling the coverage of a voxel through a continuous random variable between 0 and 1, instead of a binary representation (e.g. free or occupied), is to better model the space occupancy. For example, if a voxel is traversed by an obstacle somewhere in the middle, which divides the voxel into two halves, its coverage is likely to be 0.5, whereas it would be considered a fully occupied voxel (coverage equal to 1) if a binary representation was used.

Let

$$M_k = (\mathbf{x}_k, \mathcal{V}_k) : k \in \mathbb{N}, \tag{14}$$

be the kth batch of measurements, being $\mathbf{x}_k$ the sensor's position from where measurements are obtained and $\mathcal{V}_k$ the set of measurements belonging to the batch, provided by the robot's sensor at $t = t_k, t_k \in \mathbb{R}, k \in \mathbb{N}$. Let also

$$\mathcal{M}_k = \{M_i : i \in \mathbb{N}, i \leq k\} \tag{15}$$

be a sequence of k batches of measurements, gathered in the time interval $t_0 \leq t \leq t_k$, being t_0 the initial time before any batch of measurements. For $k = 0$ ($t = t_0$), the sequence of batches is the empty set $\mathcal{M}_0 = \emptyset$. The knowledge about the voxel's coverage C_l, after k batches of measurements, is modeled through the pdf (probabilistic belief)

$$p(c_l|\mathcal{M}_k), \quad 0 \leq c_l \leq 1. \tag{16}$$

We define the *3-D probabilistic map*, after k batches of measurements, as the set of random variables

$$\mathcal{C} = \{C_l : l \in \mathcal{Y}\}, \tag{17}$$

containing a coverage random variable for each voxel l in the 3-D discrete grid $\mathcal{Y}$. These random variables are described statistically through the set of coverage probability density functions:

$$\mathcal{P}(\mathcal{C}|\mathcal{M}_k) = \{p(c_l|\mathcal{M}_k) : l \in \mathcal{Y}\}. \tag{18}$$

The coverage of each individual voxel is assumed to be independent from the other voxels' coverage and thus $\mathcal{C}$ is a set of statistically independent random variables. The *map's joint pdf* $p(\mathcal{C}|\mathcal{M}_k)$, after k batches of measurements, can be written as

$$p(\mathcal{C}|\mathcal{M}_k) = \prod_{l \in \mathcal{Y}} p(c_l|\mathcal{M}_k). \tag{19}$$

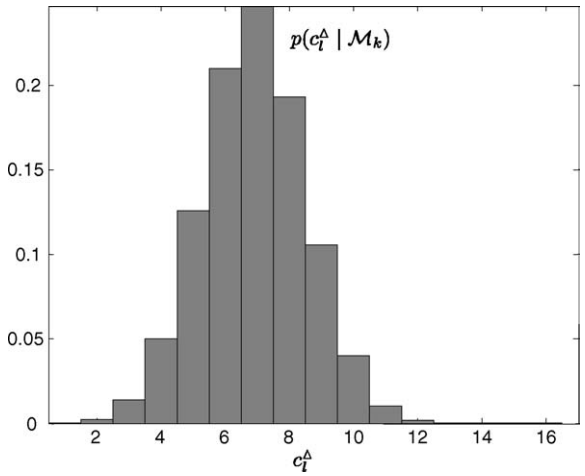

Fig. 4. Quantized version $p(c_l^{\triangle}|\mathcal{M}_k)$ of the voxel's coverage pdf $p(c_l|\mathcal{M}_k)$. The example is a 16 bins histogram built upon the pdf depicted in Fig. 3b, having $H(C_l) = 2.749$ bits.

4.2. Map's entropy

As it was mentioned in Section 3.1, the entropy definition for discrete random variables is generally preferable to differential entropy. For this reason, although we use a continuous RV to model our knowledge about the voxel's coverage, we use a quantized version of the coverage pdf to compute discrete entropy (Fig. 4). Thus, we discretise the coverage continuous RV $C_l, l \in \mathcal{Y}$ with a discrete RV $C_l^{\triangle}$ having b possible outcomes $c_l^{\triangle} \in \{1, \ldots, b\}$. This discrete RV is an approximation of the voxels' coverage pdf $p(c_l)$ through a relative frequency histogram $p(c_l^{\triangle})$ having b bins, such as:

$$p(c_l^{\triangle} = i) = \int_{\frac{i-1}{b}}^{\frac{i}{b}} p(c_l)\,\mathrm{d}c_l, \quad i \in \{1, \ldots, b\}. \tag{20}$$

Using the definition of entropy given by Eq. (1), the *voxel's entropy* is

$$H(C_l) \equiv \sum_{i=1}^{b} p(c_l^{\triangle} = i) \log p(c_l^{\triangle} = i). \tag{21}$$

Hereafter, we will always assume that we use $b = 128$ bins in the computation of Eq. (21), which means that voxel's entropy is bounded to the interval $0 \leq H(C_l) < 7$.

Since the coverage RV of different voxels are assumed to be independent, the map is a set of independent RV. Accordingly with Eqs. (12) and (13), and the map's joint pdf given by Eq. (19), the *map's joint entropy* is just the sum of voxels' individual entropy

$$H(\mathcal{C}) \equiv \sum_{l \in \mathcal{Y}} H(C_l), \tag{22}$$

which is a measure of how much uncertainty the map contains [56].

If our knowledge about the voxels' coverage is conditioned to the k previous batches of measurements $\mathcal{M}_k$, Eqs. (20)–(22) can obviously also be used to compute the voxel's coverage entropy $H(C_l|\mathcal{M}_k)$ and the map's joint entropy $H(\mathcal{C}|\mathcal{M}_k)$ conditioned to that knowledge, by using $p(c_l|\mathcal{M}_k)$ and $p(c_l^{\triangle}|\mathcal{M}_k)$ instead of using $p(c_l)p(c_l^{\triangle})$. In order to simplify our notation, the map's joint entropy $H(\mathcal{C}|\mathcal{M}_k)$ after k batches of measurements will be sometimes denoted as $H(t_k)$.

4.3. Mission execution time

Since discrete entropy is an absolute measure of uncertainty, the map's entropy given by Eq. (22) inherits that property and is an absolute measure of the map's uncertainty or quality. This property can be used to define an important performance measure, which is the mission execution time. Consider a given environment to be mapped and its associated discrete grid $\mathcal{Y}$. If different mapping missions are performed in this environment at different time periods and, perhaps, by different teams of robots, the robots' performance can be easily compared if a given map's entropy threshold H_{th} is defined. This entropy value is the minimum map's quality that robots must accomplish at the end of the mapping mission. The mission execution time $t_{k_{\max}} \in \mathcal{T}$, which is associated with the $k_{\max}$th batch of measurements, i.e. the last batch of measurements acquired by the robot with the lowest entropy at the end of the mission, can be defined as the time instant that verifies the proposition

$$H(t_{k_{\max}}) \leq H_{\text{th}} \wedge \forall_{k<k_{\max}, k \in \mathbb{N}_0}, \quad H(t_k) > H_{\text{th}}. \tag{23}$$

The mission execution time is thus the first instant time when the map's entropy is reduced below the predefined map's entropy threshold H_{th} [56]. It can be used as a performance benchmark to compare the per-

formance of different mapping missions in the same environment.

4.4. Voxels traversed by a vector

Consider an applied vector $\vec{\mathbf{u}} \in \mathbb{R}^3$ connecting point $\mathbf{a}$ to point $\mathbf{b}$ (Fig. 5a). The set of voxels traversed by $\vec{\mathbf{u}}$ can be determined by sampling it so that at least one sample per traversed voxel is gathered in a set of w 3-D points

$$\mathcal{Q}(\vec{\mathbf{u}}, \mathbf{a}) = \{\mathbf{q}_i : i \in \mathbb{N}, i \leq w\}. \tag{24}$$

To guarantee this minimum sampling, the vector $\vec{\mathbf{u}}$ is divided into segments with maximum length equal to the voxel's edge ϵ, wherein the coordinates of each sampling point are given by

$$\mathbf{q}_i = \mathbf{a} + (i-1)\epsilon\frac{\vec{\mathbf{u}}}{\|\vec{\mathbf{u}}\|}, \quad i \in \mathbb{N}, \quad i \leq w. \tag{25}$$

The number of sampled points is

$$w = \text{trunc}\left(\frac{\|\vec{\mathbf{u}}\|}{\epsilon}\right) + 1. \tag{26}$$

Let

$$v : \mathbb{R}^3 \to \mathcal{Y} \tag{27}$$

be a function which determines what grid's voxel a given point belongs to. The set of voxels traversed by vector $\vec{\mathbf{u}}$ when applied in point $\mathbf{a}$ is

$$\mathcal{Z}(\vec{\mathbf{u}}, \mathbf{a}) = \{v(\mathbf{q}_i) : \mathbf{q}_i \in \mathcal{Q}(\vec{\mathbf{u}}, \mathbf{a})\} \subset \mathcal{Y}. \tag{28}$$

4.5. Voxels influenced by a measurement

A range sensor typically provides batches of distance measurements from each point where it is located. Consider a batch of measurements $M_k = (\mathbf{x}_k, \mathcal{V}_k)$, being $\mathbf{x}_k \in \mathbb{R}^3$ the sensor's position from where measurements are obtained (shared by all measurements in the batch), and a set

$$\mathcal{V}_k = \{\vec{\mathbf{v}}_{k,i} \in \mathbb{R}^3 : i \in \mathbb{N}, i \leq m_k\} \tag{29}$$

of m_k applied vectors (measurements) connecting $\mathbf{x}_k$ to the set of points $\{\mathbf{x}_k + \vec{\mathbf{v}}_{k,i} : i \in \mathbb{N}, i \leq m_k\}$ where obstacles are detected. For each measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$, obtained from the sensor's location $\mathbf{x}_k$, we need to determine the set of voxels $\mathcal{Z}_{k,i} \subset \mathcal{Y}$ whose coverage is influenced by that measurement. As Fig. 5b suggests, this set is

$$\mathcal{Z}_{k,i} = \mathcal{Z}(\vec{\mathbf{v}}_{k,i}, \mathbf{x}_k) \cup \{l'\}, \tag{30}$$

which includes the set $\mathcal{Z}(\vec{\mathbf{v}}_{k,i}, \mathbf{x}_k)$ of voxels traversed by $\vec{\mathbf{v}}_{k,i}$ plus the voxel l' which is immediately behind

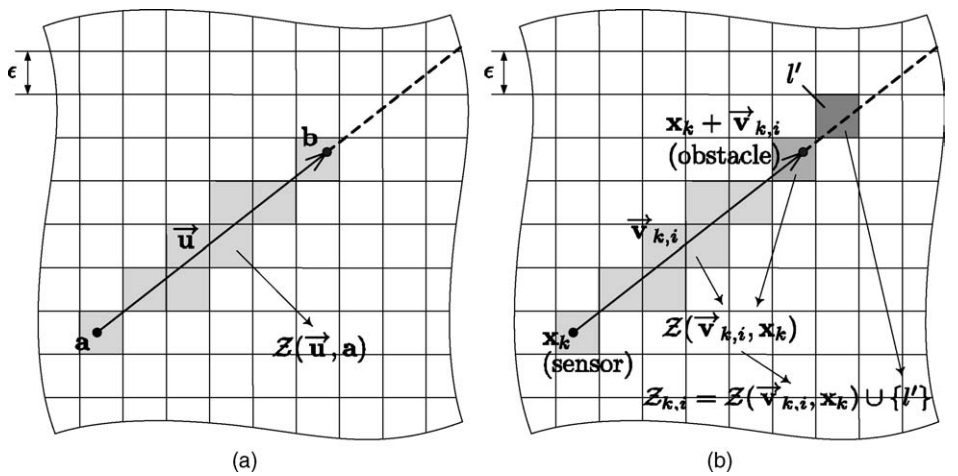

Fig. 5. Set of voxels traversed by a vector in a 2-D grid: (a) the set of traversed voxels $\mathcal{Z}(\vec{\mathbf{u}}, \mathbf{a})$ contains the shaded voxels traversed by the vector $\vec{\mathbf{u}}$ when applied in point $\mathbf{a}$ and (b) given a measurement (vector) $\vec{\mathbf{v}}_{k,i}$, when the sensor is located in the point $\mathbf{x}_k$, the set of influenced voxels $\mathcal{Z}_{k,i}$ contains the shaded voxels; light grey voxels between the sensor and the obstacle are more likely to be fully empty, dark grey voxels near to the detected obstacle have coverage values between 0 and 1, and the black voxel l', located immediately behind the detected obstacle, is more likely to be fully occupied.

the obstacle. This voxel can be computed as

$$l' = v\left(\mathbf{x}_k + \left(\text{trunc}\left(\frac{\|\vec{\mathbf{v}}_{k,i}\|}{\epsilon}\right) + 1\right)\epsilon\frac{\vec{\mathbf{v}}_{k,i}}{\|\vec{\mathbf{v}}_{k,i}\|}\right). \quad (31)$$

and is more likely to be fully occupied [56].

4.6. Measurements influencing the voxel's coverage

Consider a sensor's position $\mathbf{x}_k$ and a sensor's measurement $\vec{\mathbf{v}}_{k,i}$ obtained from that position. If $l \in \mathcal{Z}_{k,i}$, the sensor's measurement influences the coverage of the voxel $l \in \mathcal{Y}$. Let

$$\mathbf{w} : \mathcal{Y} \to \mathbb{R}^3 \quad (32)$$

be a function that computes the center coordinates $[x_l, y_l, z_l]^{\mathrm{T}}$ of a voxel $l \in \mathcal{Y}$. Let the tuple

$$D_j^l = (d_j, d_j^l) \quad (33)$$

be an individual measurement influencing the coverage estimate of voxel $l \in \mathcal{Y}$, being

$$d_j = \|\vec{\mathbf{v}}_{k,i}\| \quad (34)$$

the measured distance (distance between the sensor and the detected obstacle) and

$$d_j^l = \|(\mathbf{w}(l) - \mathbf{x}_k)\| \quad (35)$$

the distance between the sensor and the voxel's center. The set of $n_k(l)$ measurements influencing the coverage estimate of a voxel $l \in \mathcal{Y}$, after k batches of measurements, is

$$\mathcal{D}_k^l = \{D_j^l : j \in \mathbb{N}, j \le n_k(l)\} = \{D_1^l, \ldots, D_{n_k(l)}^l\}, \quad (36)$$

having cardinality

$$n_k(l) < \sum_{a=1}^{k} m_a, \quad n_k(l) \in \mathbb{N}_0, \quad (37)$$

because not all measurements yielded by the sensor necessarily influence the voxel's coverage. Given the initial empty set of influencing measurements $\mathcal{D}_0^l$, the set of influencing measurements is recursively built upon $\mathcal{M}_k$ as

$$\mathcal{D}_k^l = \mathcal{D}_{k-1}^l \cup \left[\bigcup_{i=1}^{m_k} \begin{cases} \left\{\|\vec{\mathbf{v}}_{k,i}\|, \|(\mathbf{w}(l) - \mathbf{x}_k)\|\right\}, & l \in \mathcal{Z}_{k,i} \\ \emptyset, & \text{otherwise} \end{cases}\right]. \quad (38)$$

The set $\mathcal{M}_k$ given by Eq. (15) contains all the measurements yielded by the sensor until the kth batch of measurements, but the measurements which really influence the coverage of voxel $l \in \mathcal{Y}$ are those measurements contained by the set $\mathcal{D}_k^l$. For this reason, we have the important equality

$$p(c_l|\mathcal{M}_k) = p(c_l|\mathcal{D}_k^l), \quad \forall_{l\in\mathcal{Y}, k\in\mathbb{N}_0}. \quad (39)$$

4.7. Sensor Model

The probability density function $p(c_l|D_j^l)$ represents a sensor model whereby measurements $D_j^l = (d_j, d_j^l)$ are converted in estimates of coverage values $C_l = c_l$ of a voxel l. We generally do not know the exact model of the distribution $p(c_l|D_j^l)$. However, as localization errors and sensor errors can be usually assumed to follow a Gaussian model, we represent the voxel's coverage belief through a Gaussian model

$$p(c_l|D_j^l) = N(\mu(d_j, d_j^l), \sigma(d_j, d_j^l), c_l). \quad (40)$$

wherein, accordingly with the previously defined notation, $d_j \in \mathbb{R}$ is the distance between the sensor and the detected obstacle and $d_j^l \in \mathbb{R}$ the distance between the sensor and the voxel's center [56]. Being ϵ the voxel's edge, the mean of the Gaussian is given by

$$\mu(d_j, d_j^l) = \begin{cases} 0, & (d_j^l - d_j) \le -\frac{\epsilon}{2} \\ \frac{1}{2} + \frac{d_j^l - d_j}{\epsilon}, & |d_j^l - d_j| < \frac{\epsilon}{2} \\ 1, & (d_j^l - d_j) \ge \frac{\epsilon}{2} \end{cases}. \quad (41)$$

This equation distinguishes three situations (see Fig. 5b): in the first case, the measured distance does not end in the voxel l, with $d_j^l < d_j$, and thus it is more likely that the voxel is fully empty (coverage equal to 0); in the second case, the measured distance ends in l and the mean of its coverage is inverse proportional to the amount of the voxel covered by d_j (a value between 0 and 1); in the third case, which is only applicable to the voxel l' in Eq. (30), the measured distance does not end in the voxel l, with $d_j^l > d_j$, and thus it is more likely that the voxel is fully occupied (coverage equal to 1). The standard deviation is given by

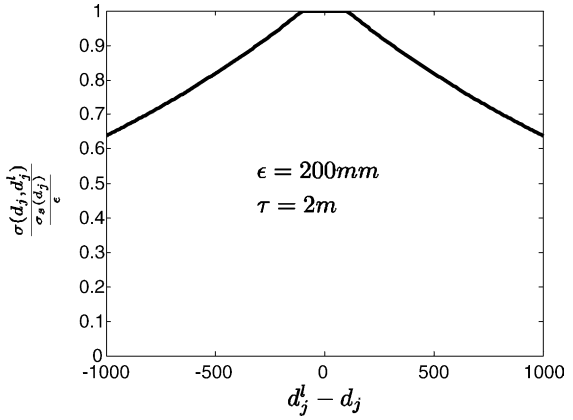

Fig. 6. Example of the coverage standard deviation damping as a function of the distance between the voxel's center and the detected obstacle.

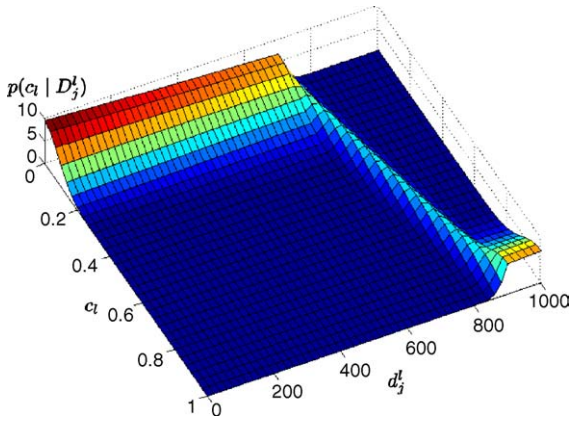

Fig. 7. Example of a sensor model: $d_j = 800\,\text{mm}$, $\sigma_{\min} = 16\,\text{mm}$, $\zeta = 1 \times 10^{-2}$, $\tau = 2\,\text{m}$, $\epsilon = 200\,\text{mm}$.

$$\sigma(d_j, d_j^l) = \begin{cases} \frac{\sigma_s(d_j)}{\epsilon}, & |d_j^l - d_j| \leq \frac{\epsilon}{2} \\ \frac{\sigma_s(d_j)}{\epsilon} \exp\left(-\frac{|d_j^l - d_j| - \frac{\epsilon}{2}}{\tau}\right), & \text{otherwise} \end{cases}, \tag{42}$$

wherein

$$\sigma_s(d) = \sigma_{\min} + \zeta d \tag{43}$$

is a linear model of the range sensor's standard deviation. It is at least equal to $\sigma_{\min}$ near to the sensor and increases linearly with distance d, which is a typical behavior of range sensors because accuracy decreases with distance. Eq. (42) states that $\sigma(d_j, d_j^l)$ is $\sigma_s(d_j)/\epsilon$ if the measured distance does end in the voxel and, given the damping ratio τ (see Fig. 6), decays exponentially with $|d_j^l - d_j|$ for voxels farther from the detected obstacle which, intuitively, have less uncertain coverage estimates.

Accordingly with the definition of coverage pdf given by Eq. (16), the Gaussian yielded by the sensor model has to be truncated so that the cumulative probability over the coverage domain sums up to one, i.e. $P(0 \leq C_l \leq 1) = 1$. In Ref. [7], it is proposed a sensor model based on a mixture of a Gaussian and an uniform distribution, wherein the latter distribution adds some white noise to ensure a correct normalization when truncating the Gaussian to the range $[0, 1]$. We claim that a better way of normalizing a normal distribution truncated to that interval is to multiply the pdf by a normalization factor

$$\gamma(\mu, \sigma) = \left(\int_0^1 N(\mu, \sigma, x)\,\mathrm{d}x\right)^{-1}, \tag{44}$$

which preserves the normal distribution instead of summing white noise [56]. Our definition of the Gaussian's mean is also slightly different from Ref. [7], for the second case of Eq. (41). Fig. 7 shows an example of the sensor model for a detected obstacle at a distance $d_j = 800\,\text{mm}$ and $d_j^l \in [0, 1000]\,\text{mm}$.

4.8. Updating the map

Updating the 3-D probabilistic coverage map (17) upon a new batch $M_k = (\mathbf{x}_k, \mathcal{V}_k)$, means updating the coverage pdf (16) of voxels influenced by the measurements contained on it. The new batch contains a set of m_k measurements $\mathcal{V}_k = \{\vec{\mathbf{v}}_{k,i} \in \mathbb{R}^3 : i \in \mathbb{N}, i \leq m_k\}$, being $\mathcal{Z}_{k,i} \subset \mathcal{Y}$ the set of influenced voxels by a measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$. Recall also that $\mathcal{D}_k^l$ is the set of influencing measurements of a voxel $l \in \mathcal{Y}$ after the kth batch of measurements, having cardinality $n_k(l)$. Let $\mathcal{D}_n^l = \{D_1^l, \ldots, D_n^l\}$ denote a set of n measurements influencing the coverage belief of a voxel $l \in \mathcal{Y}$. Note that $\mathcal{D}_k^l$ and $\mathcal{D}_{n_k(l)}^l$ are equivalent notations.

The algorithm for updating the map upon M_k can be written in pseudo-code as:

$\mathcal{D}_k^l \leftarrow \mathcal{D}_{k-1}^l, \forall_{l\in\mathcal{Y}}$
$n_k(l) \leftarrow n_{k-1}(l), \forall_{l\in\mathcal{Y}}$
for $i = 1 \ldots m_k$
  forall $l \in \mathcal{Z}_{k,i}$
    $n_k(l) \leftarrow n_k(l) + 1$
    Compute influencing measurement $D^l_{n_k(l)}$ upon $\vec{\mathbf{v}}_{k,i}$
    $\mathcal{D}_k^l \leftarrow \mathcal{D}_k^l \cup \{D^l_{n_k(l)}\}$
    Update $p(c_l|\mathcal{D}^l_{n_k(l)-1})$ upon $p(c_l|D^l_{n_k(l)})$ and obtain $p(c_l|\mathcal{D}^l_{n_k(l)})$
  end_forall
end_for.

Eqs. (34) and (35) are used to compute $D^l_{n_k(l)}$ upon a vector $\vec{\mathbf{v}}_{k,i}$ yielded by the sensor located in $\mathbf{x}_k$. The sensor model, given by Eq. (40), is used to convert sensor's measurements in new coverage estimates.

4.8.1. Initial map

The initial belief $p(c_l|\mathcal{D}_0^l) = p(c_l|\mathcal{M}_0)$ represents prior knowledge about the voxel's coverage, before any batch of measurements. Unless there is a previous map of the environment being mapped, it is usually chosen to be the less informative, i.e. a pdf with maximum uncertainty [56]. Applying the differential entropy definition given by Eq. (11) to a continuous random variable X with a Gaussian pdf $N(\mu, \sigma, x)$, it can be shown that its differential entropy increases with the logarithm of standard deviation σ as

$$h(X \to N(\mu, \sigma, x)) = \log(\sqrt{2\pi e}\sigma). \tag{45}$$

It can also be shown that Eq. (45) is a maximum entropy bound for a pdf with variance σ^2. A convenient initial belief $p(c_l|\mathcal{D}_0^l)$ is thus a Gaussian distribution with $\sigma \to +\infty$, i.e. an uniform distribution. In practice, this means choosing a Gaussian with σ much larger (e.g. 10 times greater) than the sensor standard deviation given by Eq. (42).

4.8.2. Updating the coverage belief of a voxel

Consider a given voxel $l \in \mathcal{Y}$, the set $\mathcal{D}^l_{n-1} = \{D^l_1, \ldots, D^l_{n-1}\}$ containing $n-1$ measurements influencing its coverage and its coverage belief $p(c_l|\mathcal{D}^l_{n-1})$. We are going to state how to update this belief and determine a new coverage belief $p(c_l|\mathcal{D}^l_n)$ upon a new measurement D^l_n, which is converted in a new coverage estimate through the sensor model $p(c_l|D^l_n)$, given by Eq. (40). The belief about the voxel's coverage can be computed as [56]:

$$p(c_l|\mathcal{D}^l_n) = \frac{p(\mathcal{D}^l_n|c_l)p(c_l)}{p(\mathcal{D}^l_n)} = \beta_1 p(c_l)p(\mathcal{D}^l_n|c_l) \tag{46}$$

$$p(c_l|\mathcal{D}^l_n) = \beta_1 p(c_l)\prod_{j=1}^{n} p(D^l_j|c_l) \tag{47}$$

$$p(c_l|\mathcal{D}^l_n) = \beta_1 p(c_l)\prod_{j=1}^{n} \frac{p(c_l|D^l_j)p(D^l_j)}{p(c_l)} \tag{48}$$

$$\begin{aligned} p(c_l|\mathcal{D}^l_n) &= \beta_1\beta_2 \prod_{j=1}^{n} p(c_l|D^l_j) \\ &= \beta_1\beta_2 p(c_l|D^l_n)p(c_l|\mathcal{D}^l_{n-1}). \end{aligned} \tag{49}$$

Applying Bayes rule, we obtain (46). Then, if we assume that consecutive measurements are independent given the voxel's coverage, we obtain (47). Applying again Bayes rule, we obtain (48). If we assume that $p(D^l_j)$ is constant with j, we finally obtain (49). The constants β_1 and β_2 are normalization constants ensuring that the left-hand side sums up to one over all c_l. Thus, Eq. (49) can be used recursively to update the belief $p(c_l|\mathcal{D}^l_n)$ whenever a new influencing distance D^l_n is obtained, by multiplying the current belief $p(c_l|\mathcal{D}^l_{n-1})$ with the coverage estimate $p(c_l|D^l_n)$, given the sensor model and the new influencing measurement, and applying the normalization factor $\beta_1\beta_2$. Note that for $n = 1$, i.e. for the first influencing measurement, this recursive procedure uses the initial belief $p(c_l|\mathcal{D}^l_0)$, $D^l_0 = \emptyset$.

Consider Eq. (36) giving the set of influencing measurements of voxel $l \in \mathcal{Y}$ up to the kth batch of measurements. Let $\mathcal{D}^l_{k-1} = \{D^l_1, \ldots, D^l_{n_{k-1}(l)}\}$ be the set of $n_{k-1}(l)$ measurements influencing the coverage estimate of a voxel $l \in \mathcal{Y}$ until the $(k-1)$th batch of measurements. Let $p(c_l|\mathcal{D}^l_{k-1})$ the associated voxel's coverage belief. When the sensor provides the kth batch of measurements M_k at $t = t_k$, some measurements are eventually appended to the set $\mathcal{D}^l_{k-1}$, which yields a new set of measurements

$\mathcal{D}^l_k = \{D^l_1, \ldots, D^l_{n_{k-1}(l)+1}, \ldots, D^l_{n_k(l)}\}$ having cardinality $n_k(l) > n_{k-1}(l)$. The voxel's coverage belief after the kth batch of measurements can be computed by using recursively Eq. (49) for all these new influencing measurements, as

$$p(c_l|\mathcal{D}^l_k) = \beta_3 \left[\prod_{j=n_{k-1}+1}^{n_k} p(c_l|D^l_j) \right] p(c_l|\mathcal{D}^l_{k-1}), \tag{50}$$

being β_3 a normalization constant ensuring that the left-hand side sums up to one over all c_l. Note that if $n_k(l) - n_{k-1}(l) = 0$, i.e. if there are no new influencing measurements provided by the kth batch of measurements, $\mathcal{D}^l_k = \mathcal{D}^l_{k-1}$ and, obviously, $p(c_l|\mathcal{D}^l_k) = p(c_l|\mathcal{D}^l_{k-1})$. Note also that, accordingly with the equality given by Eq. (39), we have $p(c_l|\mathcal{M}_{k-1}) = p(c_l|\mathcal{D}^l_{k-1})$ and $p(c_l|\mathcal{M}_k) = p(c_l|\mathcal{D}^l_k)$.

4.8.3. Special case of updating Gaussians

At the beginning of a mapping mission, each voxel has an associated coverage belief modeled through a Gaussian, usually having high entropy. As the sensor model (40) also yields Gaussian beliefs, Eq. (49) involves the multiplication of two Gaussians when the first influencing measurement D^l_1 is obtained. If the resultant pdf is also a Gaussian, this process repeats itself whenever new measures are gathered.

In fact, it can be easily shown that the product of two Gaussians $p(c_l|\mathcal{D}^l_{n-1}) = N(\mu_1, \sigma_1)$ and $p(c_l|D^l_n) = N(\mu_2, \sigma_2)$ yields a Gaussian multiplied by a constant [56]

$$p(c_l|\mathcal{D}^l_{n-1})p(c_l|D^l_n) = \frac{1}{\beta} N(\mu, \sigma), \tag{51}$$

$$\beta = \sqrt{2\pi(\sigma_1^2 + \sigma_2^2)} \exp\left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right], \tag{52}$$

whose parameters are given by the closed-form equations

$$\mu = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \tag{53}$$

$$\sigma = \frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}. \tag{54}$$

Comparing Eqs. (49) and (51) we conclude that: updating the coverage belief of a voxel, between consecutive influencing measurements, is as simple as computing the parameters of a new Gaussian through Eqs. (53) and (54); and the normalization constant is $\beta_1\beta_2 = \beta$, with β given by Eq. (52). This simplicity of computation is a consequence of the Gaussian nature of sensor model and our careful choice of an initial coverage belief. It is easy to conclude from Eq. (54) that the standard deviation σ of the Gaussian yielded by the product of two Gaussians with standard deviation σ_1 and σ_2, respectively, always verifies the condition $\sigma < \sigma_1 \wedge \sigma < \sigma_2$, i.e. the new voxel's coverage belief has always lower standard deviation and, accordingly with Eq. (45), lower differential entropy and uncertainty. Fig. 8 shows an example of the aforementioned update procedure. The differential entropy values of depicted pdf's are $h(C_l|\mathcal{D}^l_{n-1}) = -1.690$, $h(C_l|D^l_n) = -1.275$ and $h(C_l|\mathcal{D}^l_n) = -2.012$. The entropy values of the quantized versions are $H(C_l|\mathcal{D}^l_{n-1}) = 5.312$, $H(C_l|D^l_n) = 5.726$ and $H(C_l|\mathcal{D}^l_n) = 4.991$.

Although Gaussians' domain is not restricted to the interval [0, 1], accordingly with Eqs. (41) and (53), we can conclude that $0 \leq \mu_l \leq 1$. In practice, truncating the Gaussian to that interval is not critical to update the coverage belief but, if for some purpose we have to do

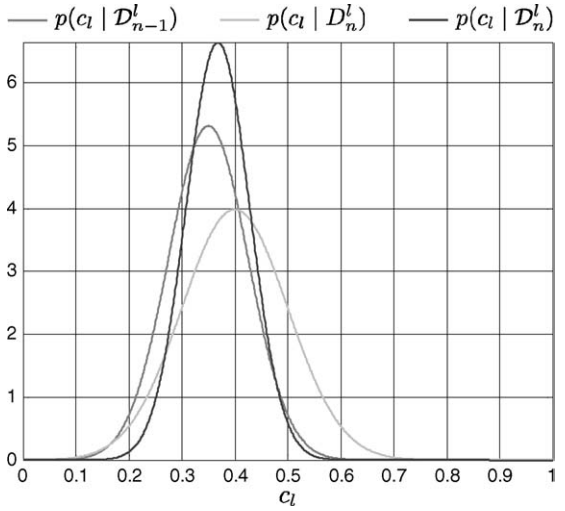

Fig. 8. Example of the coverage belief update procedure with Gaussians: $p(c_l|\mathcal{D}^l_{n-1}) = N(0.35, 0.075)$, $p(c_l|D^l_n) = N(0.4, 0.1)$ and $p(c_l|\mathcal{D}^l_n) = N(0.368, 0.06)$.

it,[3] we apply the normalization factor (44). Although the mean of the truncated Gaussian is different from μ_l, its mode is equal to μ_l and might be taken as a good estimate of the voxel's coverage, because that difference tends to zero provided that $\sigma_l \to 0$.

While in Ref. [7] the coverage belief of a cell was represented through histograms with b bins (b is typically more than 10), in our case we represent the voxel's coverage belief as a Gaussian, which is fully characterized by just *two* parameters: μ_l and σ_l. Thus, in the set of pdf given by Eq. (18), we have to store only two values for each voxel, which is a much more compact representation than a histogram. Moreover, the aforementioned procedure for updating the coverage belief is very simple and we can still build histograms upon the pdf with an arbitrary number of bins.

4.9. Entropy gradient-based exploration

In order to gradually reduce the map's entropy, the robots have to explore the environment so as to measure it from different viewpoints. In an exploration mission, the objective is to acquire as much new information about the environment as possible with every sensing cycle. When a robot has to select a new viewpoint for acquiring data through its sensor, we claim that the robot's sensor should be directed to regions having higher magnitudes of entropy gradient and low expected coverage, in the neighborhood of the robot. This strategy drives the robot's sensor to frontier voxels between more explored and less explored regions, so that the information gain of new acquired data can be maximized [56]. This strategy is a reformulation of *frontier-based exploration* [33] using the entropy concept.

4.9.1. Subset of voxels in the robot's neighborhood

Although our method can be applied to a 6 DOF robot, we have been mainly interested on using it on ground mobile robots with 3 DOF – two position coordinates x and y and orientation θ – whose sensor's motion is instantaneously restricted to a plane Γ parallel to the robot's motion plane (e.g. the floor plane in a room). For this reason, voxels near to that plane are preferable to be explored. Consider the current robot's pose $Y = (\mathbf{x}, \mathbf{a})$, being $\mathbf{x} \in \mathbb{R}^3$ its current position and $\mathbf{a} = [\theta, \phi, \psi]^{\mathrm{T}}$ its orientation. Given a robot's coordinates frame $\{R\}$, which is obtained from the global (absolute) coordinates frame $\{W\}$ after translation and rotation, the robot's motion plane Γ is defined by two orthogonal axes: a longitudinal axis $\hat{\mathbf{p}}' = [1, 0, 0]^{\mathrm{T}}$, which is the unitary vector along xx axis, and a transverse axis $\hat{\mathbf{q}}' = [0, 1, 0]^{\mathrm{T}}$, which is the unitary vector along yy axis; for example, for a UAV, $\hat{\mathbf{p}}$ would be the axis between tail and head, and $\hat{\mathbf{q}}$ would be the axis connecting the wings. It can be shown that the robot's axes can be expressed in the global coordinates frame $\{W\}$ as

$$\hat{\mathbf{p}} = [\cos\theta \cdot \cos\phi,\ \sin\theta \cdot \cos\phi,\ -\sin\phi]^{\mathrm{T}}, \tag{55}$$

$$\hat{\mathbf{q}} = \begin{bmatrix} \cos\theta \cdot \sin\phi \cdot \sin\psi - \sin\theta \cdot \cos\psi \\ \sin\theta \cdot \sin\phi \cdot \sin\psi + \cos\theta \cdot \cos\psi \\ \cos\phi \cdot \sin\psi \end{bmatrix}. \tag{56}$$

The angles θ, ϕ and ψ are the yaw angle, the pitch angle and the roll angle, respectively, and are assumed to be positive in the counterclockwise direction. Note that axis $\hat{\mathbf{p}}$ can also be viewed as the robot's sensor gaze direction. Any vector $\vec{\mathbf{u}}$ can be projected on the robot's motion plane Γ as

$$\underset{\Gamma}{\mathrm{proj}}\ \vec{\mathbf{u}} = (\vec{\mathbf{u}} \cdot \hat{\mathbf{p}})\hat{\mathbf{p}} + (\vec{\mathbf{u}} \cdot \hat{\mathbf{q}})\hat{\mathbf{q}}, \tag{57}$$

wherein $(\cdot)$ denotes the internal product of two vectors.

Let denote the applied vector connecting the robot's sensor position $\mathbf{x} \in \mathbb{R}^3$ to the center of voxel l as $\vec{\mathbf{u}}(\mathbf{x}, l) = \mathbf{w}(l) - \mathbf{x}$. Given a neighborhood around the current robot's sensor position with radius ε, its new position is selected as the center of a voxel belonging to the set of voxels

$$\mathcal{N}_\Gamma(\mathbf{x}, \varepsilon) = \{l \in \mathcal{Y},\ \|\vec{\mathbf{u}}(\mathbf{x}, l)\| \le \varepsilon,\ l = v(\underset{\Gamma}{\mathrm{proj}}\ \mathbf{w}(l))\}. \tag{58}$$

4.9.2. Entropy gradient

The 3-D grid $\mathcal{Y}$ discretises the 3-D space $\mathbb{R}^3$ at discrete points $\mathbf{w}(l), l \in \mathcal{Y}$, equally spaced by ϵ (the voxel's edge). The 3-D map enables us to associate with each of these points an entropy $H(l) = H(C_l)$ given by Eq. (21). Therefore, a continuous entropy field

[3] For instance, the purpose might be computing a cumulative probability, such as $P(0 \le C_l \le 0.3)$.

$H : \mathbb{R}^3 \to \mathbb{R}$ is sampled along the voxels' centers belonging to the grid $\mathcal{Y}$. Our volumetric model assumes that each edge of any voxel $l \in \mathcal{Y}$ is aligned with one of the axes of the global coordinates frame $\{W\}$ (see Fig. 3a). Let $l_{\Theta-}$ denote the contiguous voxel to l in the negative direction of axis Θ. A reasonable (first order) approximation to the entropy gradient at the center of a voxel l is

$$\vec{\nabla} H(l) \approx \frac{1}{\epsilon}[H(l) - H(l_{x-}), H(l) - H(l_{y-}), H(l) - H(l_{z-})]^{\mathrm{T}}. \quad (59)$$

The projection of the voxel's entropy gradient on the robot's sensor motion plane Γ is

$$\vec{\nabla} H_\Gamma(l) = \operatorname*{proj}_\Gamma \vec{\nabla} H(l), \quad (60)$$

with magnitude $\|\vec{\nabla} H_\Gamma(l)\|$.

4.9.3. Exploration strategy

Our exploration strategy drives the robot's sensor to regions in the neighborhood of the robot having higher magnitudes of entropy gradient $\vec{\nabla} H$, and that are more likely unoccupied. If the center of a voxel $l \in \mathcal{N}_\Gamma(\mathbf{x}, \varepsilon)$ is selected to be the next robot's selected position $\mathbf{x}^s$, our method selects the robot's gaze direction $\mathbf{a}(l)$, defined by the unitary vector

$$\hat{\mathbf{p}}(l) = \frac{\vec{\nabla} H_\Gamma(l)}{\|\vec{\nabla} H_\Gamma(l)\|}, \quad \vec{\nabla} H_\Gamma(l) \neq \vec{0}. \quad (61)$$

Accordingly with our exploration strategy, being $E(C_l)$ the expected coverage of a voxel $l \in \mathcal{Y}$, and given the set of voxels $\mathcal{N}_\Gamma(\mathbf{x}, \varepsilon)$ in the robot's neighborhood, the robot's sensor is directed to the voxel

$$l^s = \operatorname*{argmax}_{l \in \mathcal{N}_\Gamma(\mathbf{x}, \varepsilon)} (\|\vec{\nabla} H_\Gamma(l)\|[1 - E(C_l)]). \quad (62)$$

with a gaze on arrival defined by the unitary vector $\hat{\mathbf{p}}(l^s)$. If the gradient-based criteria is not conclusive, the robot should wander randomly until that condition is not verified.

4.10. Mobile robots equipped with stereo-vision sensors

The 3-D mapping framework presented in previous sections has been used for carrying out experiments with mobile robots in our laboratory. The mobile robots (see Fig. 9a) are Scout robots from Nomadic Technologies, having differential kinematics, odometry sensing and sonars. Each robot has inside an embedded computer running a Linux operating system, which is based on a Pentium 133 MHz processor. We mounted on the top of both robots a stereo-vision sensor and a modem radio providing wireless TCP/IP communication. Each stereo-vision sensor (see bottom of Fig. 9a) is a small, compact, low-cost analog stereo rig from Videre Design, with resolution 160×120 pixels. For computing range data from stereo images, we use the Small-Vision System (SVS) v2.3c [58], a stereo engine from SRI International, which implements an area correlation algorithm for computing range from stereo images, and supports camera calibration, 3-D reconstruction and effective filtering. See Fig. 9b for an example of a depth map yielded by the SVS engine. Each robot has a ring of 16 Polaroid 6500 sonar ranging modules, which were used for avoiding obstacles when moving the platform, and for preventing the robot to acquire stereo image pairs below a given distance threshold to obstacles. The sensor model given by Eq. (43) was properly calibrated for these stereo-vision sensors. The values $\sigma_{\min} = -0.06$ mm and $\zeta = 3.75 \times 10^{-3}$ were found. Note that our stereo-vision sensors cannot measure distances below roughly 1 m and, although $\sigma_{\min}$ might be negative, Eq. (43) always yields a positive value for $d > 1000$ mm. The damping parameter used in Eq. (42) was empirically tuned to the value $\tau = 2$ m.

4.11. Results

Fig. 10 shows Virtual Reality Modeling Language (VRML) models of a volumetric map at different instant times along a mission, which was obtained with a single robot. Each voxel is represented through a given color which depends on its coverage and its coverage belief uncertainty (entropy). The map's resolution is $\epsilon = 0.1$ m and covers a parallelepiped with 3.9 m in length, 4.2 m in width and 0.8 m in height, which was represented through a 3-D discrete grid with size $39 \times 42 \times 8$ voxels. The robot started the mission with a maximum entropy map for which $H(0) = 103.488 \times 10^3$ bits, wherein each voxel belonging to the grid had an entropy value equal to 7 bits ($b = 128$ bins for the histogram). Then, it explored gradually the environment until the map's entropy was

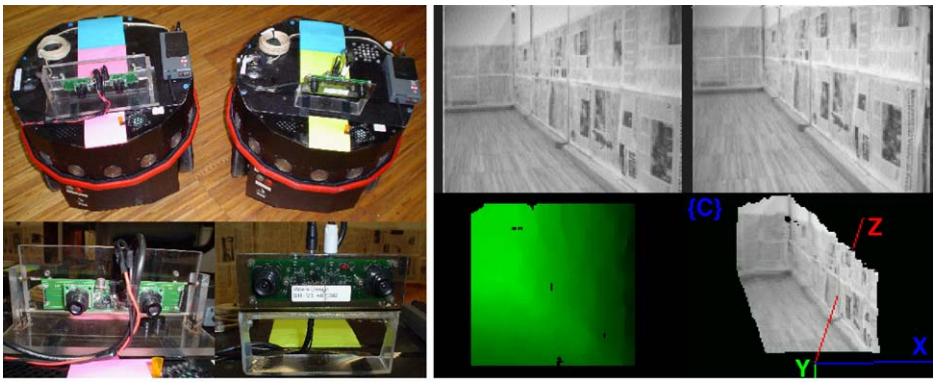

Fig. 9. Robots used in the experiments: (a) Scout mobile robots (top), equipped with stereo-vision sensors (bottom) and (b) a stereo image pair (top) and its associated disparity (bottom-left) and depth map (bottom-right).

reduced below the threshold $H_{\text{th}} = 500$ bits (see the final map on the bottom-left of Fig. 10). The robot needed $t_{k_{\max}} = 9146$ s to accomplish the mission. The total number of measurements m_{T} gathered by a robot along the mission might be computed as

$$m_{\text{T}} = \sum_{k=1}^{k_{\max}} m_k. \tag{63}$$

The robot gathered a total of $k_{\max} = 303$ batches of measurements, with an average size of $\overline{m_k} = 9049$ measurements, which yielded a total of $m_{\text{T}} = 2.742 \times 10^6$ measurements.

Fig. 11 presents the final (best) 3-D map obtained by the robot at the end of the mission, from different viewpoints.

5. Information utility

Whenever a robot gets a new batch of measurements M_k, we can say that this event has an associated information utility, which can be measured in terms of a decrease of the map's joint entropy $H(\mathcal{C})$. Let $H(t_k)$ be the map's joint entropy at $t = t_k$, computed through Eq. (22). The map's joint entropy is a measure of the map's uncertainty and its decrease within a period of time is a measure of the information utility of the measurements gathered within the same period of time, in terms of their utility on improving the map's accuracy [57].

Consider again a batch of measurements $M_k = (\mathbf{x}_k, \mathcal{V}_k)$. Each measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$ influences the coverage of the set of voxels $\mathcal{Z}_{k,i}$ and thus it has also an associated information utility. Let $l \in \mathcal{Z}_{k,i}$ be a voxel whose coverage C_l is influenced by the new measurement $\vec{\mathbf{v}}_{k,i}$; for the same voxel, let also $p(c_l|\mathcal{D}^l_{n-1}) = p(c_l|D^l_1, \ldots, D^l_{n-1})$ and $p(c_l|\mathcal{D}^l_n) = p(c_l|D^l_1, \ldots, D^l_n)$ be the coverage belief, respectively, before and after voxel l is updated with the new influencing measurement $D^l_n = (d_n, d^l_n)$, through Eq. (49). Using the conditional mutual information definition given by Eq. (10), the information utility associated with the measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$ is [57]

$$I_{k,i} = \sum_{l \in \mathcal{Z}_{k,i}} H(C_l|D^l_1, \ldots, D^l_{n-1}) - H(C_l|D^l_1, \ldots, D^l_n) = \sum_{l \in \mathcal{Z}_{k,i}} I^l_{k,i}. \tag{64}$$

Recall the definition of conditional mutual information given by Eq. (10). Each term $I^l_{k,i}$ in Eq. (64) measures the mutual information between $p(c_l|D^l_n)$ and $p(c_l)$, conditioned to the past history $\mathcal{D}^l_{n-1}$, i.e. the contribution of each influencing measurement D^l_n to reduce the voxel's uncertainty. For instance, the information utility of the influencing measurement depicted in Fig. 8 is equal to 0.321.

Although differential entropy cannot be used as an absolute measure of entropy, it is a valid relative measure of entropy, i.e. both discrete entropy and differential entropy can be used to compute the variation

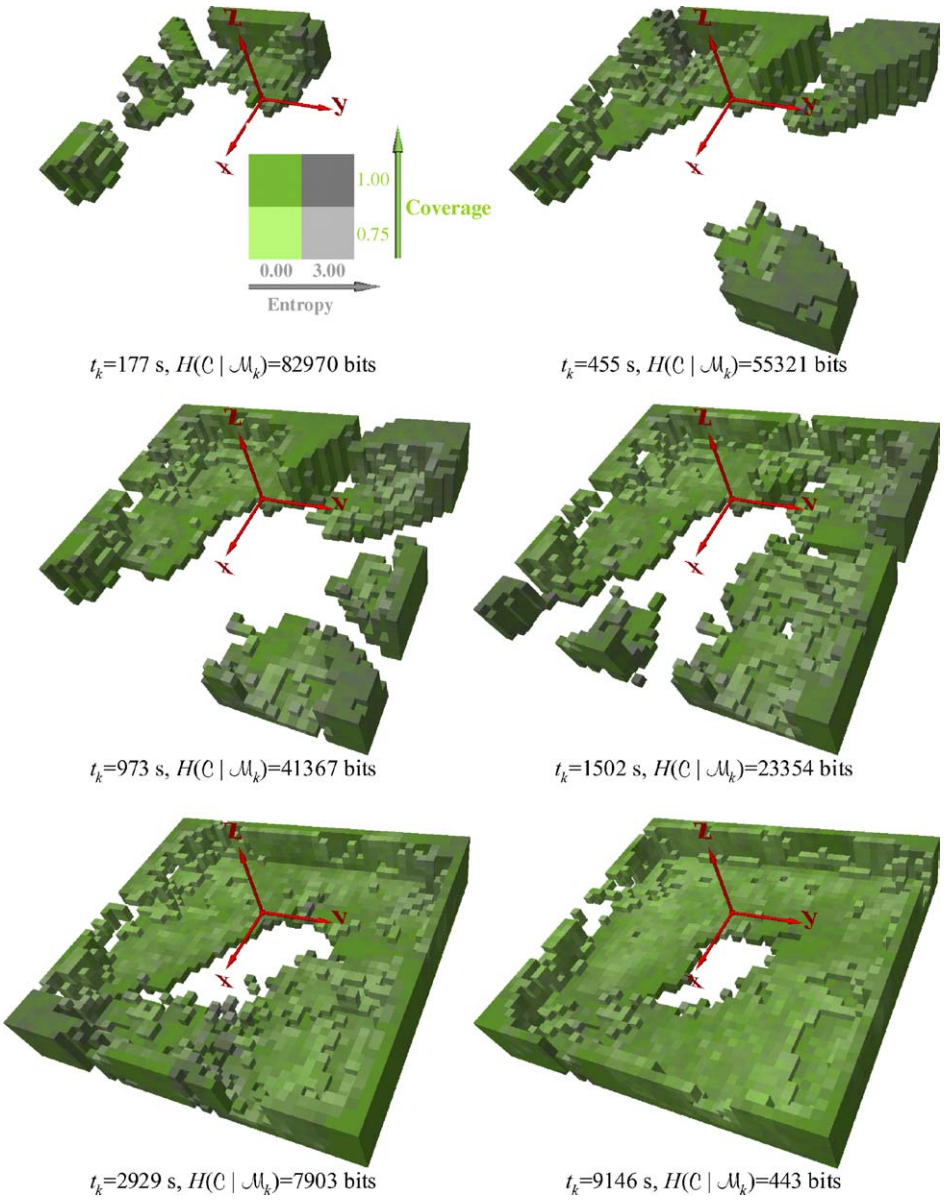

Fig. 10. Map's evolution along a 3-D mapping mission. Each snapshot shows the robot's map registered in the global reference frame $\{W\}$, at a different instant time t_k and entropy level $H(\mathcal{C}|\mathcal{M}_k)$. The map's resolution is $\epsilon = 0.1$ m. The pictures' scale is such that each represented arrow is equivalent to a real length of 1 m.

(difference) between two entropy values and this difference is equal for both. Eq. (45) provides a very convenient procedure for computing the differential entropy of a Gaussian, because it is a closed-form equation. Instead, computing the discrete entropy through Eq. (21) is computationally heavier. For each influenced voxel $l \in \mathcal{Z}_{k,i}$ in Eq. (64), let the Gaussians $p(c_l)$ and $p'(c_l)$, having standard deviation σ_l and σ'_l, be its coverage belief before and after the new measurement is integrated, having differential entropy $h(l)$ and $h'(l)$

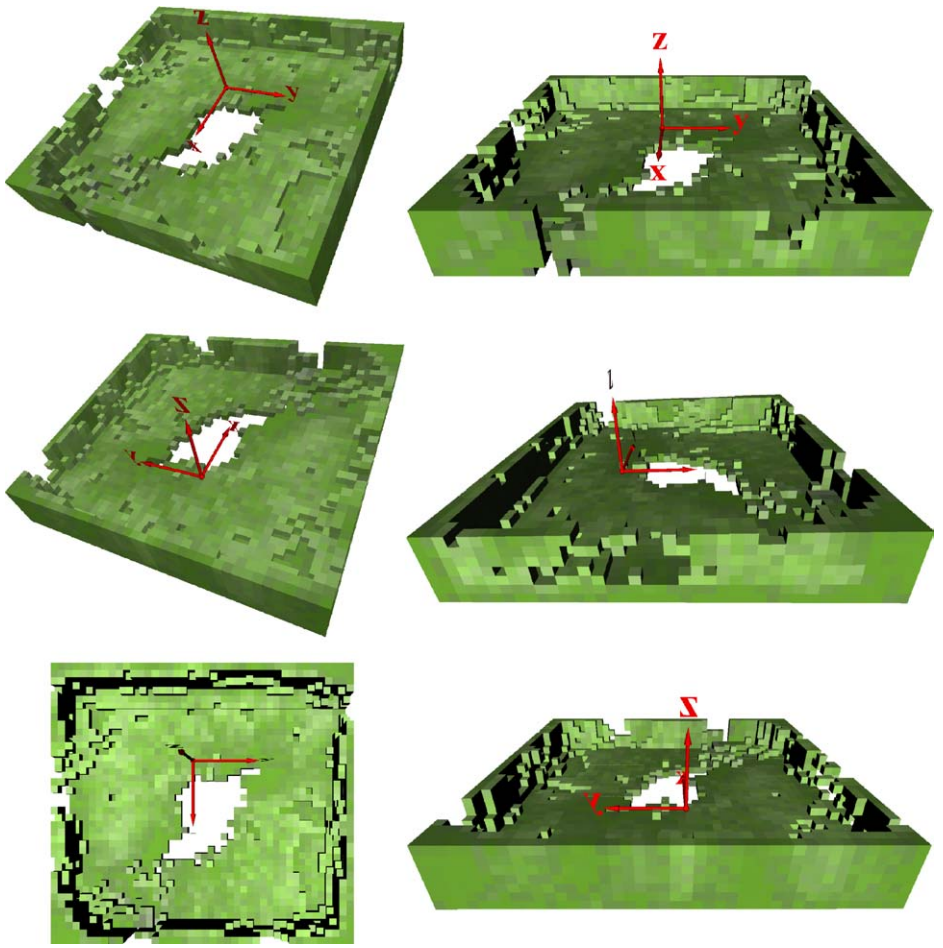

Fig. 11. Different views of the volumetric map's VRML model at the end of a 3-D mapping mission. The world reference frame is represented through the red axes.

and discrete entropy $H(l)$ and $H'(l)$, respectively.[4] Let $I^{l\delta}_{k,i} = h(l) - h'(l)$ be the measurement's information utility for that voxels, computed through Eq. (45), and $I^{l\triangle}_{k,i} = H(l) - H'(l)$ the same utility computed upon the pdf's quantized version using discrete entropy. It can be shown that

$$I^{l}_{k,i} \approx I^{l\delta}_{k,i} = \log\left(\frac{\sigma_l}{\sigma'_l}\right). \tag{65}$$

Eq. (65) states that $I^{l}_{k,i}$ is approximately equal for Gaussian pairs with the same standard deviation ratio σ_l/σ'_l. Fig. 12 compares $I^{l\triangle}_{k,i}$ and $I^{l\delta}_{k,i}$ as a function of σ_l, when $\sigma_l/\sigma'_l = 1/0.99$, i.e. when the new measurement yields a decrease of 1% in the standard deviation of the voxel's Gaussian coverage. It shows that: for $\sigma_l > 0.3$, we cannot use the differential entropy-based estimate because it neglects the pdf's truncation to the interval $0 \leq C_l \leq 1$ and the normalization introduced by Eq. (44); and, for $\sigma_l < 3 \times 10^{-3}$, we cannot use the discrete entropy-based estimate, because the histogram's bins have not sufficient resolution to model pdf's with a smaller standard deviation. However, for $3 \times 10^{-3} \leq \sigma_l \leq 0.3$, which encompasses most of the situations with our stereo-vision sensors, both estimates are approximately equivalent. This conclusion

[4] Recall that we compute discrete entropy of a continuous pdf through a histogram derived from it having $b = 128$ bins.

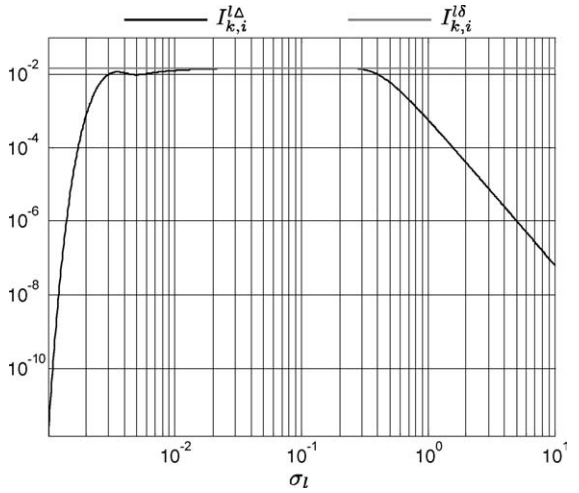

Fig. 12. Computation of the information utility $I^l_{k,i}$ of a measurement for a given influenced voxel l. Being σ_l and $\sigma'_l = 0.99\sigma_l$ the standard deviation of the voxel's coverage Gaussian before and after the measurement, respectively, the graph plots the computed values of $I^l_{k,i}$ as a function of σ_l, using either the histogram's discrete entropy $I^{l\Delta}_{k,i}$ (histogram with $b = 128$ bins) or the approximated voxel's coverage differential entropy $I^{l\delta}_{k,i} = 1.45 \times 10^{-2}$.

is roughly the same for other values of the ratio σ_l/σ'_l, though it is not explicitly shown herein due to lack of space. For this reason, we generally use Eq. (65) in the computation of Eq. (64).

The information gain due to the kth batch of measurements is given by

$$I_k = \sum_{i=1}^{m_k} I_{k,i} = H(t_{k-1}) - H(t_k), \tag{66}$$

which measures the mutual information between the current map and the new acquired batch of measurements, i.e. the contribution of this batch to reduce the map's uncertainty. As Eq. (22) requires the computation of the entropy function for *every* voxel $l \in \mathcal{Y}$, it represents a time-consuming computation if it is used at each time step, whenever a new batch of measurements is gathered. But, as Eq. (66) suggests, the map's joint entropy can be recursively updated as $H(t_k) = H(t_{k-1}) - I_k$, which is a much more efficient computation procedure because it is only computed the Eq. (64) for each measurement $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$ belonging to the batch $M_k = (\mathbf{x}_k, \mathcal{V}_k)$. Thus, Eq. (22) is only required for computing the maps's initial entropy $H(0) = H(\mathcal{C}|\mathcal{M}_0)$.

5.1. Cooperation through sharing useful information

In this section, we describe how a team of such robots, populating the 3-D environment being mapped, can be cooperative on building a 3-D map, through sharing *useful* measurements. Accordingly with the architecture model shown in Figs. 1 and 2, we assume that, besides being able to build and update its own local 3-D map based on information from its own sensor, each robot is also committed to share new acquired sensory information with its teammates through communication [57]. Whenever a given robot gets a batch of measurements $M_k = (\mathbf{x}_k, \mathcal{V}_k)$, it sends to other robots a sub-set of measurements $S_k = (\mathbf{x}_k, \mathcal{U}_k)$. The set

$$\mathcal{U}_k = \{\vec{\mathbf{u}}_{k,1}, \ldots, \vec{\mathbf{u}}_{k,s_k}\} \subseteq \mathcal{V}_k \tag{67}$$

contains s_k measurements selected to be communicated. The sensor's position $\mathbf{x}_k$ from where those measurements were gathered is also sent, since it is required for registering those measurements in the local map of other robots.[5]

Different communication topologies can be used, depending on the capacity and range of the available communication channel. When possible, the robot acting as information provider should send data to all robots in the team, so that all of them can take advantage of new sensory information; otherwise, the communication is restricted to a team's subset, such as the nearest robots to the information provider. When a robot receives a batch of u_k communicated measurements $R_k = (\mathbf{x}'_k, \mathcal{U}'_k)$, it updates its local map as if measurements $\mathcal{U}'_k$ would have been gathered by its own sensor when located at position $\mathbf{x}'_k$.

As communication channels have always limited capacity, when a robot is acting as information provider, it has to limit the amount of communicated data and select the most useful measurements gathered from its own sensors. On doing it, the robot uses Eq. (64) to assess the information utility of measurements $\vec{\mathbf{v}}_{k,i} \in \mathcal{V}_k$ and classifies them by utility. Let define $s_{k_{\max}}$ as being the maximum number of allowable communicated measurements at a given time instant. Let also define $I_{\min}$ as being the minimum allowable information utility for a communicated measurement. The set (67) is

[5] As it was already mentioned, we assume that each robot is able to localize itself and correctly register its sensor measurements.

built in such a way that the proposition

$$(s_k \leq s_{k_{\max}} \wedge s_k < s_{k_{\max}} \Rightarrow \forall_{\vec{\mathbf{v}}_{k,z} \in \mathcal{V}_k \setminus \mathcal{U}_k}, I_{k,z} < I_{\min} \wedge \forall_{\vec{\mathbf{u}}_{k,j} \in \mathcal{U}_k}, I_{k,j} \geq I_{\min} \wedge \forall_{\vec{\mathbf{v}}_{k,w} \in \mathcal{V}_k \setminus \mathcal{U}_k}, I_{k,w} \leq I_{k,j}) \quad (68)$$

is true. This proposition is true, i.e. the set of communicated measurements is valid, if the following conditions are met: (a) the size of the set is not greater than $s_{k_{\max}}$; (b) the size of the set is less than $s_{k_{\max}}$ only if it includes all measurements in the set $\mathcal{V}_k$ having an information utility not less than $I_{\min}$; (c) the information utility of communicated measurements is at least $I_{\min}$ and all not communicated measurements have lower or equal utility than those that are selected to be communicated.

Accordingly with Eq. (64), the sender robot assesses the measurement's utility by assuming that if the measurement is useful for itself it is equally useful for its teammates. Although different robots may have different maps, as we shall see in Section 5.3, typically the robots' maps are just slightly different and, thus, the assumption remains valid.

5.2. Implementation in mobile robots

The robots presented in Section 4.10 were used for carrying out cooperative 3-D mapping experiments, using the architecture model shown in Fig. 1 and the cooperation scheme described in Section 5.1 for sharing *useful* sensory data [57]. Fig. 13 shows a diagram of the software that was programmed on the robots in order to implement the distributed architecture model of Fig. 1. It is intrinsically scalable to a team having an arbitrary number of robots.

Fig. 13a shows the interaction between the team of robots and a host PC, which was used for supervision. Using this computer, the user can control the mission execution through the module `MASTERCTR` (e.g. start, pause, restore, stop, etc.), get access to robots' data through the module `MAPCOLLECT` (e.g. robots' individual maps, log data, etc.), and visualize VRML models of the maps through the module `VISUALIZ` (see some examples in Fig. 11). The host PC is also responsible for providing global localization to the team of robots through the module `LOCALIZSRV`, using a color segmentation algorithm that detects and tracks

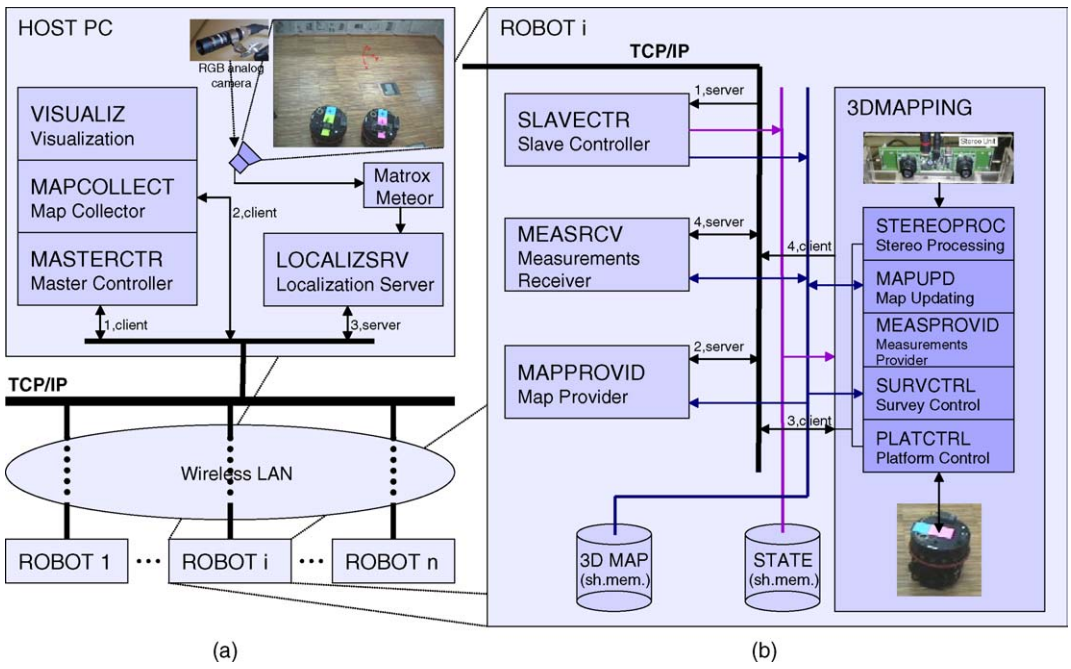

Fig. 13. Diagram of the software implemented in mobile robots: (a) interaction between the team of robots and the host PC used for supervising 3-D mapping missions and localizing the robots through a global camera and (b) software modules running locally on each robot.

the colored markers on the top of the robots' platforms (see Fig. 9a).

Each robot is able to build autonomously a volumetric map on its own, storing and maintaining locally the grid-based map on a shared memory area, which is shared by the different modules running concurrently in the robot (see Fig. 13b). The mutual exclusive access to the map's shared memory is ruled through a semaphor provided by the Linux kernel. There is also a shared memory area for storing and sharing the robot's state among those modules. Each robot runs locally three software modules: `SLAVECTR`—receives commands from the module `MASTERCTR` running remotely in the host PC, which determine the current robot's state; `MAPPROVID`—sends incrementally data to to the host PC's module `MAPCOLLECT` during the mission execution, so as to synchronize the map's copy stored in the host PC with the robot's own copy, whenever the robot updates the map upon new measurements; `3DMAPPING`—performs most of the robot's computation burden, including acquiring stereo image pairs, computing range data from stereo images, updating the map upon new measurements, selecting new exploration viewpoints and controlling the motion of the robot's platform.

5.3. Results and discussion

This section presents results obtained within experiments carried out with a team of two cooperative mobile robots (see Fig. 9a), which were programmed accordingly with the software architecture shown in Fig. 13. These experiments aimed at studying the influence of the information sharing parameters $I_{\min}$ and $s_{k_{\max}}$ on the team's performance, by comparing the mission execution time $t_{k_{\max}}$ with different values for those parameters [57]. As in the experiments reported in Section 4.11, the robots started each experiment with a maximum entropy map and used the entropy gradient-based method described in Section 4.9 for exploring the environment until the entropy threshold $H_{\text{th}} = 500$ was attained.

The environment, the initial map and the stopping criteria were fixed for all the experiments. Table 1 summarizes the obtained results with the team of two robots, which are however extensible and can be generalized to teams having an arbitrary number of robots, because the robots' program is intrinsically scalable to any team size. The fourth column shows the ratio between the mission execution time $t_{k_{\max}}(2)$ with two robots and with one robot $t_{k_{\max}}(1)$. Given that our voxels' coverage beliefs were always Gaussians, the values used for $I_{\min}$, {0, 0.00723, 0.01450, 0.07400, 0.15200, 0.32193}, meant an average reduction on the standard deviation of the influenced voxels by a measurement of at least {0%, 0.5%, 1%, 5%, 10%, 20%}, respectively. Recall that when a robot acquired a new batch of m_k measurements $M_k = (\mathbf{x}_k, \mathcal{V}_k)$ through its own sensor, it might sent to the other robot a batch of s_k useful measurements $S_k = (\mathbf{x}_k, \mathcal{U}_k)$, with $s_k \leq m_k \wedge s_k \leq s_{k_{\max}} \wedge u_k = 0$. Conversely, a kth batch of m_k measurements $M_k = (\mathbf{x}_k, \mathcal{V}_k)$ might not be acquired from its own sensor and thus might be a batch of u_k useful measurements sent (shared) by the other robot $M_k = R_k = (\mathbf{x}'_k, \mathcal{U}'_k)$, with $u_k = m_k \leq s_{k_{\max}} \wedge s_k = 0$. The fifth column shows the total number of measurements m_{T} gathered by a robot along the mission, which is given by Eq. (63). The sixth column shows the total number of received measurements from the other robot u_{T}, which is computed through

$$u_{\text{T}} = \sum_{k=1}^{k_{\max}} u_k. \tag{69}$$

In each experiment (line of the table), the results refer to the robot that first attained the entropy threshold H_{th}, i.e. the robot having the best map at the end of the mission. Fig. 14 presents an example of the maps obtained by the two robots along a 3-D mapping mission. As we can observe, robot 2 held the best map for the instant times represented in the figure, which means that robot 2 reached H_{th} first. The time $t_k(1)$ that a single robot would need to obtain the represented maps is also shown, so as to better understand the reduction of the mission execution time yielded by a team of cooperative mobile robots.

5.3.1. Advantages provided by cooperation

The graph on the left of Fig. 15 compares the map's entropy $H(t)$ for the single robot case and for the fastest experiment with two robots (fourth row of Table 1). It shows a non-linear increase of the mission execution time with a decrease of the map's entropy. It also shows that robots' cooperation accelerated the reduction of the map's entropy and led to a reduction of

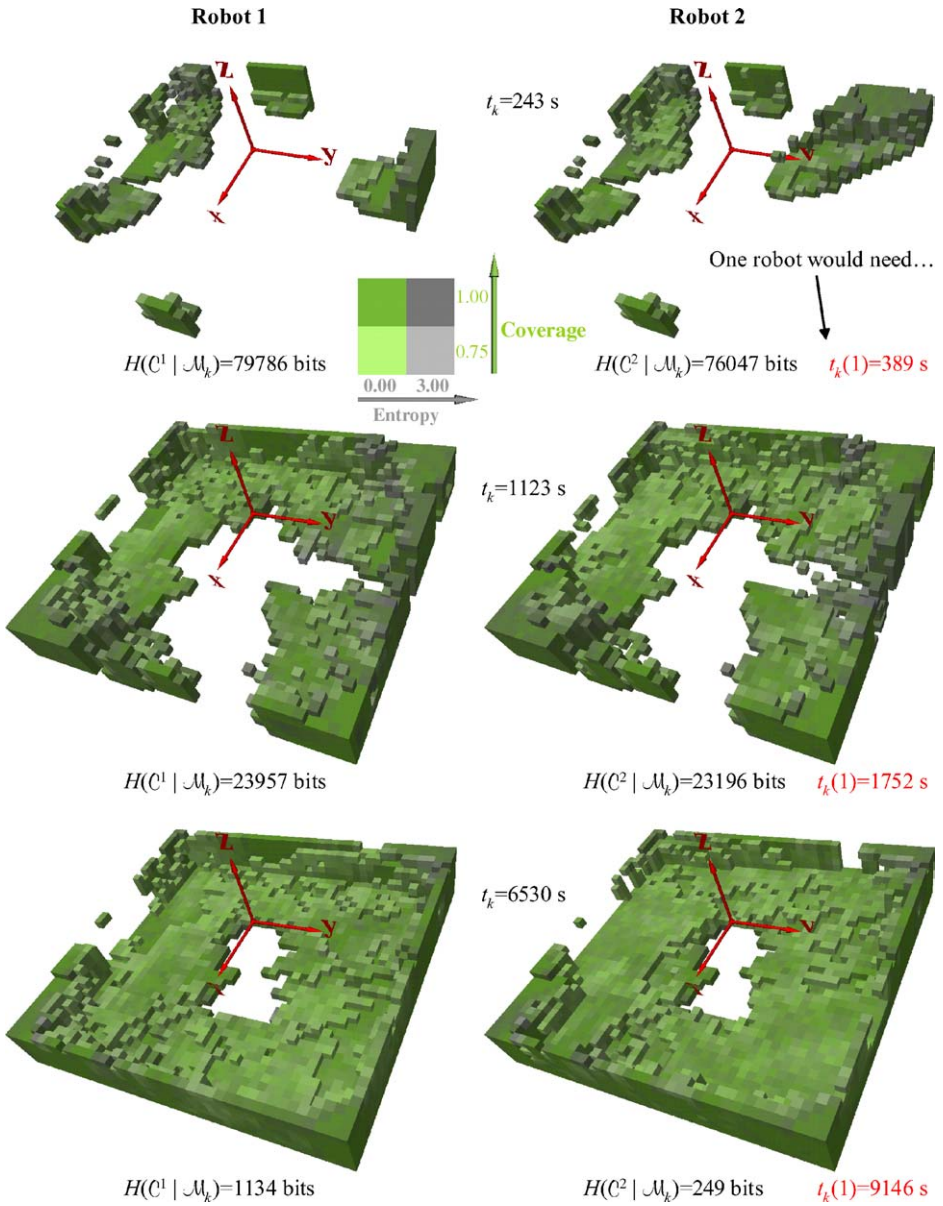

Fig. 14. Maps' evolution along a 3-D mapping mission with two robots. Each row shows a snapshot of the map of each robot registered in the global reference frame $\{W\}$, at a different instant time t_k and entropy level $H(\mathcal{C}|\mathcal{M}_k)$. In every rows, the best map was held by robot 2 (maps shown on the right). The time $t_k(1)$ that a single robot would need to obtain a map with the same entropy is shown on the bottom-right of the maps of robot 2. The map's resolution is $\epsilon = 0.1$ m. The pictures' scale is such that each represented arrow is equivalent to a real length of 1 m. For the presented case, the parameters $s_{k_{\max}} = 2500$ and $I_{\min} = 0.1520$ were used.

28% in $t_{k_{\max}}$. As robots shared useful measurements through communication, each robot was able to integrate in its map a greater number of measurements per time unit and achieved a faster reduction of its map's entropy. The graph on the right of Fig. 15 shows that although the two values of m_{T} were similar, measurements were obtained within time intervals $t_{k_{\max}}$ quite different.

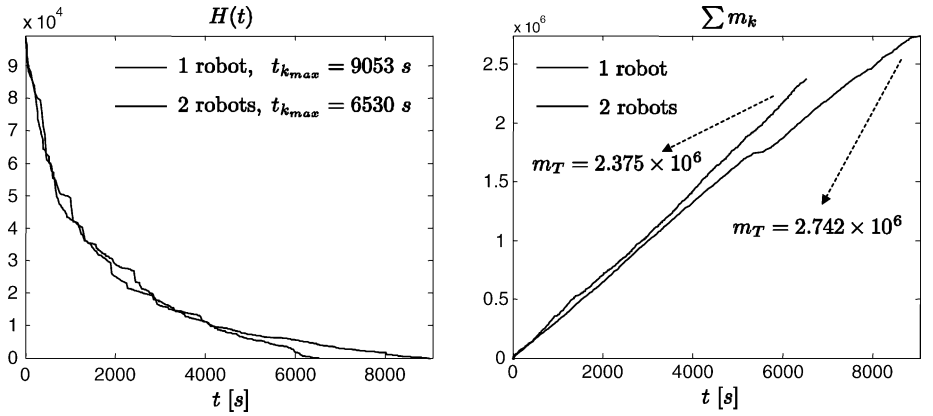

Fig. 15. Comparison of a 3-D mapping mission using a single robot or two robots: entropy of the map along the mission (left) and cumulative number of processed measurements along the mission (right). For the presented case, the parameters' values $I_{\min} = 0.0145$ and $s_{k_{\max}} = 2500$ were used.

Besides enabling cooperation and its aforementioned benefits, the coexistence of several robots in the same workspace and the communication among robots also yield some pitfalls contributing for the degradation of the team's overall performance: (a) robots must share the workspace which leads to some mutual interference; (b) the time spent on communicating measurements to other robots sometimes delays operations; (c) the time required for processing received measurements through communication and updating the map upon them might not be negligible. While the two latter problems depend mostly on the communication bandwidth and the robots' computation power, minimizing the interference among robots is an interesting future extension of the framework proposed herein.

Table 1
Results obtained within experiments with two robots and different parameters ruling the information sharing

$s_{k_{\max}}$	$I_{\min}$	$t_{k_{\max}}$	$\frac{t_{k_{\max}}(2)}{t_{k_{\max}}(1)}$	m_T	u_T	Percent
500	0.01450	8483	0.94	2795351	74729	3
1000	0.01450	8387	0.93	2726837	135661	5
1750	0.01450	7332	0.81	2447091	184550	8
2500	0.01450	6530	0.72	2375273	207636	9
5000	0.01450	7955	0.88	2643728	271612	10
20000	0	9450	1.04	3192788	1134455	36
20000	0.00723	7563	0.84	2453021	457390	19
20000	0.01450	6571	0.73	2345844	332270	14
20000	0.07400	7007	0.77	2676612	128345	5
20000	0.15200	7301	0.81	2595398	59499	2
20000	0.32193	7727	0.85	2930155	27323	1

5.3.2. Influence of communication selectivity

Fig. 16 presents a graph of $t_{k_{\max}}$ as a function of the parameter $I_{\min}$ and curves of the cumulative sum $\sum u_k$ of received measurements from the other robot along the mission, for different values of $I_{\min}$. For this figure, the maximum number of allowable communicated measurements at a given time instant was $s_{k_{\max}} = 20{,}000$. Since the number of measurements yielded by the sensor was about 10^4 measurements, in this situation, $s_{k_{\max}}$ did not restrict the communication for any acquired batch of measurements, because $m_k < s_{k_{\max}}$, $1 \leq k \leq k_{\max}$. The graph on the left of Fig. 16 shows that decreasing $I_{\min}$ from 0.32193 to 0.01450 led to smaller mission execution times. However, for $I_{\min} < 0.01450$, the graph of $t_{k_{\max}}$ presents a remarkable inflection, which led to a fast degradation of the team's performance. This observation puts on evidence the importance of selecting the most useful information to be communicated. If the selection is too weak, most of the communicated information becomes redundant and the time spent on communicating and processing that superfluous information becomes very significant [57]. The curves on the right of Fig. 16 show that the first derivative is the same at the beginning of the mission, because $s_{k_{\max}}$ is common to all of them. However, as long as the mission is executed, the derivative decreases to an extent which depends on the selectivity introduced by $I_{\min}$.

The graph on the right of Fig. 17 shows that reducing the communication bandwidth $s_{k_{\max}}$ always led to

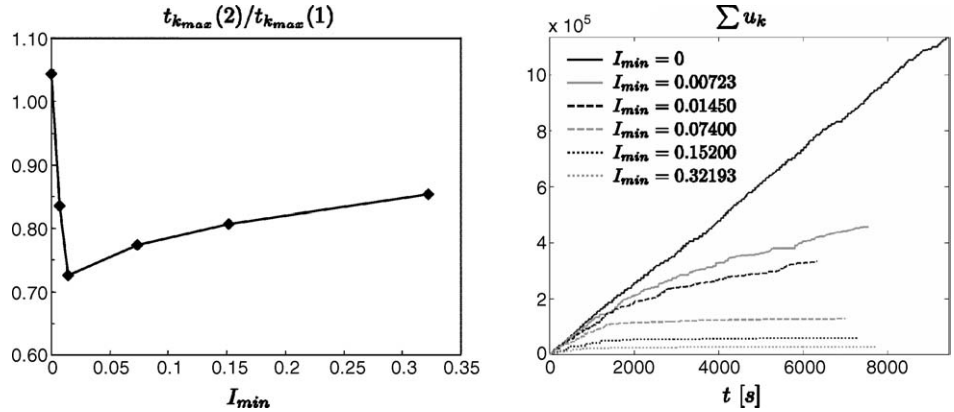

Fig. 16. Influence of the information selectivity – parameter $I_{\min}$ – on the mission execution time with two robots, using $s_{k_{\max}} = 20{,}000$: mission execution time (left) and cumulative number of received measurements from the other robot along the mission (right).

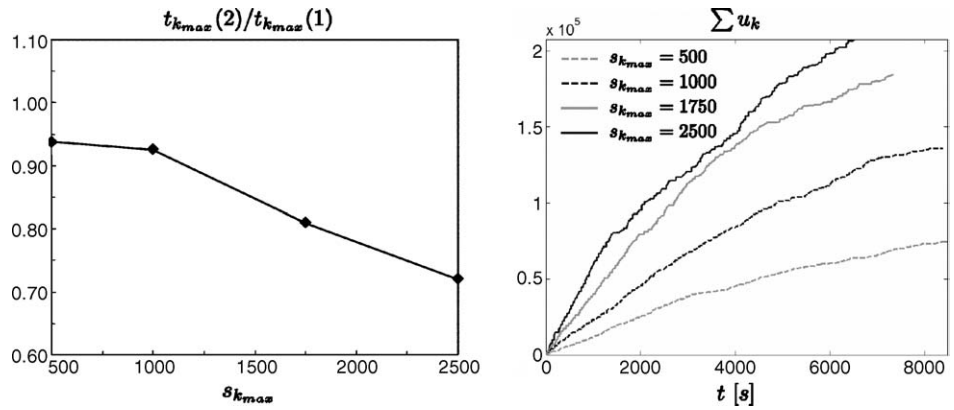

Fig. 17. Influence of the maximum number of communicated measurements – parameter $s_{k_{\max}}$ – on the mission execution time with two robots, using $I_{\min} = 0.01450$: mission execution time (left) and cumulative number of received measurements from the other robot along the mission (right).

an increase of $t_{k_{\max}}$ and a poorer team's performance [57]. As cooperation in a 3-D mapping mission relies completely on explicit communication, restricting it also restricts the extent of cooperation. However, in the case of $I_{\min}$, being selective to some extent is beneficial in order to select the most useful information and to avoid the communication of redundant information. The curves on the right of Fig. 17 show that the first derivative by the end of the mission is the same, because $I_{\min}$ is common to all of them; and that it is as high as $s_{k_{\max}}$ at the start, since this parameter configures a bandwidth limitation.

These results yield some guidelines for developing communication schemes aiming at fostering cooperation [57]. While $s_{k_{\max}}$ is imposed by the communication channel capacity, $I_{\min}$ has to be tuned in an intelligent way, whereby its selective power is beneficial for the robotic team's performance. It should be selective enough to avoid communicating redundant information, and not too selective so as to enable efficient information sharing and cooperation among robots.

6. Conclusion

Most of the previous research on building maps has been restricted to models without an explicit represen-

tation of the map's uncertainty. This article presents innovative work related with developing a probabilistic model for 3-D mapping, by using information theory to formally represent uncertainty. The advantages of this probabilistic approach include: a very compact representation of 3-D grid-based coverage maps; a straightforward method for updating the map upon new range measurements, which is based on a Bayes filter; and an entropy-based formulation of frontier-based exploration, which was demonstrated to converge nicely to maps with lower uncertainty.

Furthermore, a distributed architecture model for a team of cooperative mobile robots was proposed, whose main feature is to foster cooperation through sharing useful information. With this purpose, a formal entropy-based measure of information utility was proposed in order to endow robots with the ability of selecting purposively useful information. Experimental results demonstrated that teams of cooperative robots are able to build a map in less time than a single robot.

Possible future extensions of the approach presented in this article include a method to coordinate the exploration with multiple robots, a hierarchy of maps' representations and the integration of cooperative localization schemes. The former extension would take maximum advantage from the cooperation among robots, yielding a more significant speedup from one robot to two robots, by minimizing their interference (e.g. avoiding the situation wherein more than one robot sense the same volume). The latter two extensions would both increase the robustness of the approach, especially in outdoor, large environments, where sometimes global localization schemes are either unreliable or not viable. A hierarchy of representations would take advantage from the complementary advantages of metric and topological representations, so as to be able to represent with high resolution local maps of more interesting regions, while being able to represent huge volumes through a topological representation, whose nodes are metric maps. Cooperative localization would take advantage from multi-robot localization methods, which have been demonstrated to be more accurate than single robot localization methods [24]. Using relative observations, different robots can refine their internal beliefs based on the other robots' estimate and improve localization accuracy.

References

[1] Y. Cao, A. Fukunaga, A. Kahng, Cooperative mobile robotics: antecedents and directions, Auton. Robots 4 (1997) 1–23.

[2] G. Dudek, M. Jenkin, E. Milios, A taxonomy of multirobot systems, in: T. Balch, L. Parker (Eds.), Robot Teams: From Diversity to Polymorphism, A.K. Peters Ltd., 2002.

[3] T. Arai, E. Pagello, L. Parker, Special issue on advances in multirobot systems, IEEE Trans. Rob. Autom. 18 (5) (2002) 655–864.

[4] A. Farinelli, L. Iocchi, D. Nardi, Multirobot systems: a classification focused on coordination, IEEE Trans. Syst. Man Cybernet. B 34 (5) (2004) 2015–2028.

[5] S. Thrun, Robotic mapping: a survey, in: G. Lakemeyer, B. Nebel (Eds.), Exploring Artificial Intellelligence in New Millenium, M. Kaufmann, 2002.

[6] M. Martin, H. Moravec, Robot evidence grids, Technical Report CMU-RI-TR-96-06, Robotics Institute, CMU, Pittsburgh, USA, 1996.

[7] C. Stachniss, W. Burgard, Mapping and exploration with mobile robots using coverage maps, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003), 2003, pp. 467–472.

[8] R. Madhavan, G. Dissanayake, H. Durrant-Whyte, Map-building and map-based localization in an underground-mine by statistical pattern matching, Proceedings of International Conference on Pattern Recognition, 1998.

[9] D. Huber, N. Vandapel, Automatic 3D underground mine mapping, Proceedings of the Fourth International Conference on Field and Service Robotics, July 2003.

[10] S. Thrun, D. Hahnel, D. Ferguson, M. Montermelo, R. Riebwel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, W. Whittaker, A system for volumetric robotic mapping of underground mines, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2003), 2003.

[11] M. Maimone, L. Matthies, J. Osborn, E. Rollins, J. Teza, S. Thayer, A photo-realistic 3-D mapping system for extreme nuclear environments: Chernobyl, in: Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'98), vol. 3, 1998, pp. 1521–1527.

[12] H. Moravec, A. Elfes, High resolution maps from wide angle sonar, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'85), 1985.

[13] D. Pagac, E. Nebot, H. Durrant-Whyte, An evidential approach to map-building for autonomous vehicles, IEEE Trans. Rob. Autom. 14 (4) (1998) 623–629.

[14] J. Borenstein, Y. Koren, The vector field histogram: fast obstacle avoidance for mobile robots, IEEE J. Rob. Autom. 7 (3) (1991) 278–288.

[15] B. Grocholsky, A. Makarenko, H. Durrant-Whyte, Information-theoretic coordinated control of multiple sensor platforms, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003), 2003, pp. 1521–1526.

[16] E. Hygounenc, I. Jung, P. Soueres, S. Lacroix, The autonomous blimp project of LAAS-CNRS: achievements in flight control and terrain mapping, Int. J. Rob. Res. 23 (2004) 473–511.

[17] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Trans. Rob. Autom. 17 (3) (2001) 229–241.
[18] D. Hähnel, W. Burgard, D. Fox, S. Thrun, An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003), 2003.
[19] J. Folkesson, P. Jensfelt, H. Christensen, Vision SLAM in the measurement subspace, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), 2005, pp. 30–35.
[20] V. Nguyen, A. Martinelli, R. Siegwart, Handling the inconsistency of relative map filter, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), 2005, pp. 651–656.
[21] E. Brunskill, N. Roy, SLAM using incremental probabilistic PCA and dimensionality reduction, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), 2005, pp. 344–349.
[22] C. Stachniss, G. Grisetti, W. Burgard, Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), 2005, pp. 667–672.
[23] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artif. Intell. J. 128 (1–2) (2001) 99–141.
[24] D. Fox, W. Burgard, H. Kruppa, S. Thrun, A probabilistic approach to collaborative multi-robot localization, Auton. Robots 8 (3) (2000) 325–344.
[25] S. Roumeliotis, G. Bekey, Distributed multirobot localization, IEEE Trans. Rob. Autom. 18 (5) (2002) 781–795.
[26] S. Roumeliotis, I. Rekleitis, Propagation of uncertainty in cooperative multirobot localization: analysis and experimental results, Auton. Robots 17 (1) (2004) 41–54.
[27] A. Martinelli, F. Pont, R. Siegwart, Multi-robot localization using relative observations, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), 2005, pp. 2808–2813.
[28] S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots, Int. J. Rob. Res. 20 (5) (2001) 335–363.
[29] A. Mourikis, S. Roumeliotis, Performance bounds for cooperative simultaneous localization and mapping (C-SLAM), Proceedings of Robotics: Science and Systems, Massachusetts Institute of Technology, U.S.A., June 8–11, 2005.
[30] A. Howard, Multi-robot simultaneous localization and mapping using acausal particle filters, Proceedings of Robotics: Science and Systems, Massachusetts Institute of Technology, U.S.A., June 8–11, 2005.
[31] V. Sujan, S. Dubowsky, T. Huntsberger, H. Aghazarian, Y. Cheng, P. Schenker, An architecture for distributed environment sensing with application to robotic cliff exploration, Auton. Robots 16 (2004) 287–311.
[32] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, H. Durrant-Whyte, Information based adaptive robotic exploration, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2002), 2002, pp. 540–545.
[33] B. Yamauchi, Frontier-based exploration using multiple robots, Proceedings of Second International Conference on Autonomous Agents, 1998, pp. 47–53.
[34] W. Burgard, M. Moors, D. Fox, R. Simmons, S. Thrun, Collaborative multi-robot exploration, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2000), vol. 1, 2000, pp. 476–481.
[35] W. Burgard, D. Fox, S. Thrun, Active mobile robot localization by entropy minimization, Proceedings of Second Euromicro Workshop on Advanced Mobile Robots (EUROBOT'97), 1997.
[36] J. Ko, B. Stewart, D. Fox, K. Konolige, B. Limketkai, A practical, decision-theoretic approach to multi-robot mapping and exploration, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003), 2003, pp. 3232–3238.
[37] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schutz, B. Stewart, R. Vincet, Centibots: very large scale distributed robotic teams, Proceedings of International Symposium on Experimental Robotics, Singapore, 2004.
[38] A. Howard, L. Parker, G. Sukhatme, The SDR experience: experiments with a large-scale heterogeneous mobile robot team, Proceedings of International Symposium on Experimental Robotics, Singapore, 2004.
[39] C. Stachniss, D. Hahnel, W. Burgard, Exploration with active loop-closing for FastSLAM, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2004), 2004.
[40] L. Parker, ALLIANCE: an architecture for fault-tolerant multirobot cooperation, IEEE Trans. Rob. Autom. 14 (2) (1998) 220–240.
[41] D. Jung, An architecture for cooperation among autonomous agents, Ph.D. Thesis, Department of Computer Science, University of Wollongong, Australia, 1998.
[42] M. Matarić, G. Sukhatme, Task-allocation and coordination of multiple robots to planetary exploration, Proceedings of 10th International Conference on Advanced Robotics, 2001.
[43] B. Gerkey, M. Matarić, Sold!: auction methods for multirobot coordination, IEEE Trans. Rob. Autom. 18 (October) (2002) 758–768.
[44] R. Brooks, A robust layered control system for a mobile robot, IEEE J. Rob. Autom., RA-2 (1986) 14–23.
[45] R. Arkin, Motor schema based mobile robot navigation, Int. J. Rob. Res. 8 (4) (1989) 92–112.
[46] R. Arkin, Cooperation without communication: multi-agent schema based robot navigation, J. Rob. Syst. 9 (3) (1992) 351–364.
[47] M. Matarić, Minimizing complexity in controlling a mobile robot population, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'92), 1992, pp. 830–835.
[48] T. Balch, R. Arkin, Communication in reactive multiagent robotic systems, Auton. Robots 1 (1) (1994) 27–52.

[49] T. Fukuda, K. Sekiyama, Communication reduction with risk estimate for multiple robotic system, Proceedings of IEEE International Conference on Robotics and Automation, 1994, pp. 2864–2869.
[50] L. Parker, The effect of action recognition and robot awareness in cooperative robotic teams, Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'95), 1995, pp. 212–219.
[51] M. Tambe, Towards flexible teamwork, J. Artif. Intell. Res. 7 (1997) 83–124.
[52] P. Stone, M. Veloso, Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork, Artif. Intell. 110 (2) (1999) 241–273.
[53] C.E. Shannon, The Mathematical Theory of Communication, University of Illinois Press, 1949.
[54] T. Cover, J. Thomas, Elements of Information Theory, John Wiley & Sons, 1991.
[55] A. Makarenko, S. Williams, H. Durrant-Whyte, Decentralized certainty grid maps, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2003), 2003, pp. 3258–3263.
[56] R. Rocha, J. Dias, A. Carvalho, Exploring information theory for vision-based volumetric mapping, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2005), Edmonton, Canada, August, 2005, pp. 2409–2414.
[57] R. Rocha, J. Dias, A. Carvalho, Cooperative multi-robot systems for vision-based 3-D mapping using information theory, Proceedings of IEEE International Conference on Robotics and Automation (ICRA'2005), Barcelona, Spain, April, 2005, pp. 386–391.
[58] K. Konolige, D. Beymer, SRI Small Vision System User's Manual, SRI International, Software Version 2.3, April 2002.

Rui Rocha was born on 13 May 1973, in Castelo de Paiva, north of Portugal. He completed his electrical and computer engineering degree (specialization on automation, control and instrumentation) in July 1996 and his MSc degree (specialization on industrial informatics) in March 1999, by the Faculty of Engineering of the University of Porto. Since February 2000, he has been a teaching assistant at the Department of Electrical and Computer Engineering and a researcher at the Institute of Systems and Robotics, in the Faculty of Sciences and Technology of the University of Coimbra, where he is currently working towards his PhD degree. His main research topics are cooperative multi-robot systems, distributed architectures and discrete-event dynamic systems.

Jorge Dias was born on 7 March 1960, in Coimbra, Portugal. He completed his electrical engineering degree (specialization on computers) by the Faculty of Sciences and Technology from the University of Coimbra in July 1984 and PhD degree on electrical engineering by the University of Coimbra, specialization in control and instrumentation, November 1994. He has been a teacher at the Department of Electrical and Computers Engineering, in the Faculty of Sciences and Technology of University of Coimbra, where he is currently associate professor. His main research area is computer vision and robotics, with activities and contributions on the field since 1984. His current research is carried out at the Institute of Systems and Robotics, University of Coimbra.

Adriano Carvalho was born on 30 September 1953, in Vila Nova de Famalicão, north of Portugal. He completed his electrical engineering degree in July 1971 and his PhD degree in electrical engineering in July 1989, by the Faculty of Engineering of the University of Porto. Since 1976, he has been a teacher at the Department of Electrical and Computers Engineering, in the Faculty of Engineering of the University of Porto, where he is currently associate professor. His main research area is control and automation, with activities and contributions for the past 2 decades.