International Journal of Automation and Computing 15(3), June 2018, 310-324
DOI: 10.1007/s11633-017-1109-4

Iterated Conditional Modes to Solve Simultaneous Localization and Mapping in Markov Random Fields Context

J. Gimenez A. Amicarelli J. M. Toibero F. di Sciascio R. Carelli

Automatics Institute, National University of San Juan, San Juan 5400, Argentina

Abstract: This paper models the complex simultaneous localization and mapping (SLAM) problem through a very flexible Markov random field and then solves it by using the iterated conditional modes algorithm. Markovian models allow to incorporate: any motion model; any observation model regardless of the type of sensor being chosen; prior information of the map through a map model; maps of diverse natures; sensor fusion weighted according to the accuracy. On the other hand, the iterated conditional modes algorithm is a probabilistic optimizer widely used for image processing which has not yet been used to solve the SLAM problem. This iterative solver has theoretical convergence regardless of the Markov random field chosen to model. Its initialization can be performed on-line and improved by parallel iterations whenever deemed appropriate. It can be used as a post-processing methodology if it is initialized with estimates obtained from another SLAM solver. The applied methodology can be easily implemented in other versions of the SLAM problem, such as the multi-robot version or the SLAM with dynamic environment. Simulations and real experiments show the flexibility and the excellent results of this proposal.

Keywords: Simultaneous localization and mapping, Markov random fields, iterated conditional modes, modelling, on-line solver.

1 Introduction

1.1 The problem

Simultaneous localization and mapping (SLAM) is one of the basic problems in mobile robotics. SLAM is the dual process of building a consistent and incremental map of the previously unexplored environment and using this map to ascertain the robot absolute pose or even the path traveled by the robot[1–3]. In order to build a map from the environment, the vehicle must be equipped with a sensory system capable of taking measurements of the relative location between landmarks and the vehicle itself.

SLAM applications now exist in a wide variety of domains. They include indoor[4], outdoor[5], aerial[6] and subsea[7, 8]. Several vehicle architectures have been used: unmanned aerial vehicle (UAV)[9, 10], differential drive mobile robots (DDMR)[11, 12], or car-like vehicles[13]. There are different sensing modalities such as sonar[14], range lasers[15], cameras[16, 17], global positioning systems (GPS)[18] or combinations thereof[13]. There exist generalizations of the SLAM problem such as SLAM with dynamic objects[19] or the multiple-robot SLAM problem[20, 21]. Some recent publications tackle the problem by using multiple maps, or sub-maps that are lately used to build a larger global map[22].

The interdependence between localization and mapping converts to the SLAM in a complex problem that requires the precise resolution of two problems at the same time. Speed and consistency in the resolution are the two main issues in the SLAM research community[23, 24]. The main sources of uncertainty in SLAM are three: sensor noise, association data and the accumulated errors during the navigation.

The robot sensors are classified into exteroceptive and proprioceptive. Among the exteroceptive sensors, it is possible to find: sonar, range lasers, cameras and GPS. All of these sensors are noisy and have limited range capabilities. In addition, only local views of the environment can be obtained using the first three aforementioned sensors. On the other hand, GPS data are not always available.

Proprioceptive sensors, such as encoders and inertial measurement units, allow the robot to obtain velocity, position changes and acceleration measurements. Encoders, for instance, allow obtaining an incremental estimate of the vehicle movements by means of a dead reckoning navigation method (also known as deduced-reckoning), but due to cumulative errors, they are not precise enough to have an accurate estimation of the position all the time.

Research Article
Manuscript received April 22, 2017; accepted October 25, 2017; published online February 1, 2018
This work was supported by the National Council for Scientific and Technological Research (CONICET) and the National University of San Juan (UNSJ).
Recommended by Associate Editor Nazim Mir-Nasiri

The most fundamental key topic related to all SLAM solutions is the data association problem, which arises when landmarks and observations cannot be uniquely matched, and due to this, the number of possible associations may grow exponentially, making the SLAM absolutely unviable for large areas[25]. Traditional methodologies assume that the matching problem is solved by an external procedure and it is not explicitly incorporated in the estimation process.

Recently, a methodology based on random finite sets (RFSs) was introduced, which incorporates the matching problem in the optimization process circumventing the necessity for any fragile data association and map management heuristics[26]. The article solves the SLAM problem by using the Bayesian set-based estimator PHD (probability hypothesis density) filter. However, its implementation is computationally expensive.

Navigation errors are caused by the inevitable divergence between the assumed motion model for the robot and its actual motion. This divergence can lead to an accumulative error in the estimated pose of the robot as well as aggravate the data association problem.

Then, the correct resolution of the SLAM problem requires a model that allows efficiently to incorporate sensory data of any kind, to optimize matchings observation-object, and to incorporate complex motion models (dynamic or highly nonlinear) for the vehicle.

State-of-the-art approaches to SLAM typically formulate the inference problem as a sparse high-dimensional nonconvex M-estimation[27], and then apply general first or second order smooth optimization methods to estimate a critical point of the objective function. The papers[1, 3] provide a survey of classical methods to solve the SLAM problem with its advantages and disadvantages.

Solutions to the SLAM problem involve finding an appropriate representation for both the observation model and motion model[1, 2]. By far, the most common representation is in the form of a state-space model with additive Gaussian noise, leading to the use of the extended Kalman filter (EKF) to solve the SLAM problem[28, 29]. One important alternative representation is to describe the vehicle motion model as a set of samples of a more general non-Gaussian probability distribution. This leads to the use of the Rao-blackwellized particle filter, or FastSLAM algorithm[30], to solve the SLAM problem. An improved FastSLAM algorithm based on revised genetic resampling and square root unscented particle filter (SR-UPF) is presented in [31]. While EKF-SLAM and FastSLAM are the two most important solution methods, newer alternatives, which offer much potential, have been proposed in the graphical model context[32–35].

The optimization methods that solve the SLAM problem are usually only able to guarantee convergence to a first-order critical point of the objective (i.e, local minimum or saddle point), rather than the globally optimal solution[36]. The most serious limitation is that the solution to which any such method converges is determined by its initialization. This is particularly damaging in the context of SLAM, in which the combination of a high-dimensional state space and significant nonlinearities in the objective function can give rise to complex cost surfaces containing many local minima in which smooth optimization methods can become entrapped[37]. Together, these uncertainties culminate into a complex SLAM problem, which some behaviorists believe is not worth solving[38].

1.2 Related works

Graph theory allows intuitively illustrating the states of a model with their probabilistic interdependencies. A node is assigned to each state and correlations are expressed in terms of direct (edges) and indirect (paths) dependencies. For more details on graph theory and probabilistic models defined on graphs, see [39].

Folkesson and Christensen introduced the GraphSLAM system which finds the best robot trajectory using a nonlinear optimization technique[40]. Dellaert and Kaess exploited the inherent sparsity of the SLAM problem to make the process more efficient[33]. For this, the problem is formulated through a belief net representation, which is a directed acyclic graph that encodes the conditional independence structure of a set of variables, where each variable directly depends only on their predecessors in the graph[39]. The SLAM problem is solved by optimizing the posterior probability with the large and sparse standard least-squares method. This requires a specific methodology, which is called square root SAM (square root smoothing and mapping information) and is based on an optimization process that reduces edges to optimize the amount of calculations. In [41], the SLAM problem is modeled through a directed acyclic graph and various techniques are exposed in order to solve it. Unlike previous work, the observations and control actions are incorporated as model states in the graph.

On the other hand, in [32, 42], the SLAM problem is modeled through undirected graph, i.e., the interconnections between the states are bidirectional without a causal order. Topologically, a node for the robot pose at each sampling period and a node for each landmark are generated. Then, these nodes are linked according to whether in certain sampling period a landmark is observed or not. Later, potential functions are defined according to assumptions of Gaussianity (and linearity in the case of [32]) of the motion and observation models. Finally, a procedure for solving the problem SLAM is used to optimize topologically the graph in question avoiding unfeasible computational costs.

Dissanayake et al.[43] showed that it is possible to remove a large percentage of the landmarks from the map without making the map building process statistically inconsistent. Since then many works tried to reduce the graph complexity[44–48].

A random field is Markovian (MRF) if the conditional probabilities of the model are consistent with the topological structure of the graph[49]. The potentiality of working in the MRF context has been reported in [33]. Also, Jadaliha and Choi[34] investigate a fully Bayesian way to solve the simultaneous localization and spatial prediction (SLAP) problem by using a Gaussian Markov random field (GMRF) model with satisfactory results.

1.3 Proposal

From the beginning of the century, the SLAM problem is studied through graph models[32, 33, 41]. Similarly, over the last several decades, these Markovian models have been used in the image processing context, and an endless number of useful tools to solve various problems have been developed[50]. One of them, is the iterated conditional modes (ICM) methodology[51], which has not yet been implemented for the SLAM problem. ICM is used to find the state configuration that, given the observations, locally maximizes the posterior probability of a Markovian model.

This paper proposes to model the SLAM problem through a very flexible MRF, that allows to incorporate an observation model (regardless of the sensor type used and the nature of the data acquired), a motion model (regardless of the robotic architecture), a map model (incorporating any possible definition of map) and that allows the sensory fusion preserving theoretical convergence, among other benefits. Then, the SLAM problem is solved through the particularization of ICM to the SLAM context. ICM is an iterative algorithm that is initialized with an estimate of the map and the robot trajectory. Next, iteration-by-iteration, corrects these estimates until it converges to a sub-optimal map and a sub-optimal trajectory. These sub-optimal results are better than the initial estimates, and therefore, it can be used to improve the output of any SLAM procedure. ICM has good potential to work with initializations of intermediate quality, but its results are better if it is correctly initialized. In this paper, we propose an ICM initialization that can be run on-line and improved through ICM iterations performed at appropriate times. Therefore, the main contributions of this paper are firstly the proposition of ICM to solve the SLAM problem modelled through an MRF and secondly its on-line associated initialization.

In Section 2, the necessary concepts of MRF theory and ICM to develop this paper are formally defined. In Section 3, the SLAM problem is modeled in the MRF context, and it is solved by using ICM. In Section 4, an example of application is shown, which is simulated in Section 5 and is experimented in a real environment in Section 6. Conclusions are given in Section 7.

2 Markov random fields

2.1 Definitions

Given a finite set V, whose elements are called nodes, a random field (RF) is an array $\boldsymbol{X} = \{X_v : v \in V\}$, where each X_v is a random variable (RV) that takes values in a set of possible states Ω_v. Thus, $\Omega = \prod_{v\in V}\Omega_v$ is the set of all the possible realizations of the RF$\boldsymbol{X}$.

Let X_A be the RF composed by the RVs X_v, with $v \in A \subset V$. In the particular case that $A = V - \{v\}$, it will be denoted with X_{-v}. With $\omega_A \in \prod_{v\in A}\Omega_v$, it is denoted the entry subset of $\omega \in \prod_{v\in V}\Omega_v$ defined over A.

Let P be the probability defined on Ω, where $P(\omega)$ is the probability that the RF$\boldsymbol{X}$ takes the value ω. In principle, each variable X_v depends probabilistically on all other RV. This can be reduced through the neighborhood concept between nodes, which determines the radius of conditional dependence of each variable. That is, for each node $v \in V$, a non-empty neighborhood $\partial v \subset V$ is defined, so that $v \notin \partial v$ and $v' \in \partial v \Leftrightarrow v \in \partial v'$ (see Fig. 1).

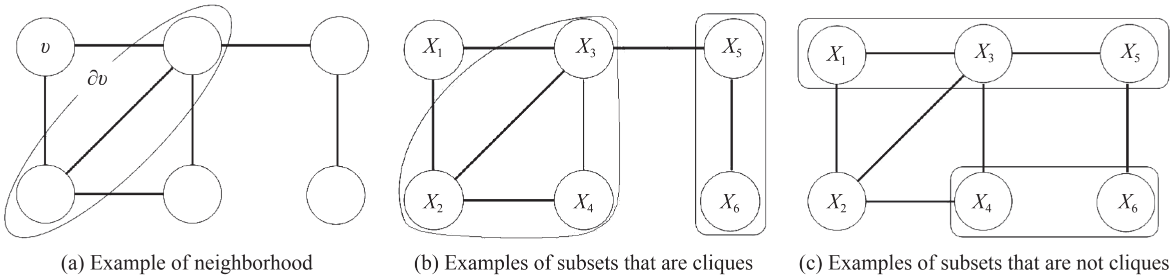
In this way, a graph $G = (V, \mathcal{E})$ is defined, where $\mathcal{E} \subseteq V \times V$ is the set of edges composed of all pairs of neighboring nodes.

A random field $\boldsymbol{X}$ with associated probability P and defined on a graph G is a MRF (Markov random field) if

$$P(\omega_v|\omega_{-v}) = P(\omega_v|\omega_{\partial v}), \quad \forall v \in V.$$

That is, if the conditional probability of observing ω_v given the value in the rest of the nodes can be calculated with only knowing the state of their neighbors.

A subset $C \subseteq V$ is a clique if $\#C = 1$ or if all pairs of nodes in C are neighbors (see Figs. 1(b) and 1(c)). Let $\boldsymbol{C}$ be the set of all the cliques of the graph. Given $C \in \boldsymbol{C}$,



(a) Example of neighborhood (b) Examples of subsets that are cliques (c) Examples of subsets that are not cliques

Fig. 1 Diagram of topological concepts defined

according to the problem to be modeled, there are configurations ω_C that are observed more frequently and therefore should be enhanced or privileged over other configurations. To this end, and in order to define the probability P that best explains the problem to be modeled, a family of potentials is defined $\Phi = \{\Phi_C : C \in \boldsymbol{C}\}$, where each $\Phi_C : \prod_{v\in C}\Omega_v \to \mathbf{R}$ is a function which verifies that $\Phi_C(\omega_C) > \Phi_C(\omega'_C)$ when the configuration ω_C is more common than the configuration ω'_C.

The function of total energy of the model $H : \Omega \to \mathbf{R}$ is defined as

$$H(\omega) = -\sum_{C\in\boldsymbol{C}} \Phi_C(\omega_C).$$

Note that energy is lower in the most common configurations of the model.

Then, from the Theorem of Hammersley-Clifford[52,49], the probability or Gibbs measure P over Ω associated to the MRF $\boldsymbol{X}$ is

$$P(\omega) = \frac{1}{Z}\exp\left\{\frac{-H(\omega)}{T}\right\} \tag{1}$$

where $Z = \sum_{\omega\in\Omega}\exp\left\{\frac{-H(\omega)}{T}\right\}$ is a normalizing constant and $T > 0$ is a constant called system temperature. Note that the lower the temperature, the greater the gap between the probability of desirable configurations over undesirable configurations. Strictly speaking, the temperature should be incorporated from the potential definitions and should not appear out of nowhere in the Gibbs measure. However, for clarity in future derivations, it is desirable that T appears explicitly in (1).

Then, from (1) and algebraic steps, it can be proved that the conditional probabilities are

$$P(\omega_v|\omega_{-v}) \propto \exp\left\{\frac{1}{T}\sum_{C\in\boldsymbol{C}:v\in C}\Phi_C(\omega_C)\right\} \tag{2}$$

where ω_C is composed of the corresponding entries of ω_v and ω_{-v}.

2.2 Gibbs sampler and simulated annealing

Since $\#\Omega$ is generally very large, direct sampling methods of the MRF $\boldsymbol{X}$ turn out to be expensive computationally. Therefore, it must be resorted to iterative methods which have the theoretical property of converging to a realization of $\boldsymbol{X}$. In practice, since this theoretical convergence is asymptotic, a reasonable number of iterations are performed. This number depends on the model in question and the final configuration is considered a realization of $\boldsymbol{X}$.

Gibbs sampler (GS) is the most used iterative sampler of Gibbs measures[53]. GS is applicable when the joint distribution is explicitly unknown or it is difficult to sample from the joint distribution directly, but the conditional distribution of each variable is known and is easy (or at least, easier) to sample from. The GS algorithm updates the state of each variable according to its conditional probability given the current values of the other variables. It can be shown (see, i.e., [54]) that the sequence of samples constitutes a Markov chain, whose stationary distribution is just the desired joint distribution. Gibbs sampling is particularly well-adapted to sampling the posterior distribution of a Bayesian network, since Bayesian networks are typically specified as a collection of conditional distributions.

Then, the general GS algorithm is summarized as

Algorithm 1. Gibbs sampler

1) Choose arbitrarily an initial configuration of states $\omega^{(0)} \in \Omega$.

2) Generate a visit scheme, i.e., to sort arbitrarily the elements of V.

3) Perform a sweep, i.e., to visit the nodes in the order specified in 2), and to replace randomly the actual state ω_v by a state of Ω_v according to the probabilities $P(\cdot|\omega_{-v})$.

4) Repeat 3) as often as wished.

This procedure converges asymptotically to a realization of $\boldsymbol{X}$[54]. In many problems, such as SLAM, the objective is to find the state configurations $\omega \in \Omega$ that maximizes $P(\omega)$ instead of sampling the Gibbs distribution.

Then, if the system temperature is decreased in each iteration of GS according to a particular cooling scheme, it is expected that the algorithm converges to a maximum of P. This is because the probability of the mode of $P(\cdot|\omega_{-v})$ is increasingly prevalent over the other elements of Ω_v. This iterative procedure is called simulated annealing (SA), and its theoretical convergence to a global maximum of P is proved if the cooling scheme satisfies a certain bound[55, 56]. The ideal cooling rate cannot be determined beforehand, and should be adjusted for each particular application. Thus, SA convergence could be very slow and several algorithms were proposed to optimize its computational cost.

This lock on the cooling scheme produces a very slow convergence of SA, and therefore there exists in the literature several algorithms that seek to optimize their computational cost. The method is an adaptation of the Metropolis-Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system.

2.3 Iterated conditional modes

In statistics, iterated conditional modes (ICM)[51] is a deterministic algorithm for obtaining the configuration that locally maximizes the joint probability of an MRF. It operates by iteratively maximizing the probability of each variable conditioned on the rest. Note that the cooling scheme increases the probability to choose the state

ω_v that maximizes $P(\omega_v|\omega_{-v})$ in each visit of SA, whereas the ICM algorithm directly chooses the mode of $P(\cdot|\omega_{-v})$. Then, the convergence of ICM to a global maximum of P may not happen. However, ICM converges to the local maximum of P closest to the initial configuration with a computational cost and a rate much slower than that required by SA. In many applications, this local maximum gives very satisfactory results if the initial configuration is properly chosen. In Section 3, the SLAM problem is presented in the MRF context and its solution by using ICM is shown.

3 Simultaneous localization and mapping

Consider a vehicle whose state at each discretized time $t = 0, 1, \cdots, T_f$ is completely determined by a vector x_t. In this paper, it is assumed that x_0 is known. If this was not so, the final estimates of the map and the robot path turn out to be a translation and rotation of the actual results, which is logical if it does not have other prior knowledge of the environment. We denote with u_t the control actions applied to the robot at time $t = 0, 1, \cdots, T_f$. In general, the control actions are the motor torques, or their velocities. If the robot has a disturbance-free movement and a perfect tracking of the control actions, then there is a relationship functional $x_t = g(x_{t-1}, u_{t-1})$ that predicts its future state for all possible state and control action of the previous time. However, this is unreal, and only it is usually expected that $g(x_{t-1}, u_{t-1})$ is an estimation of x_t.

Suppose that in the operation space of the vehicle, there are L landmarks with locations m_l, where $l = 1, 2, \cdots, L$. Consider that the robot is equipped with sensors capable of detecting landmarks in an operating range (dependent on sensor characteristics), and that at each time $t = 0, 1, \cdots, T_f$, this information can be synthesized in a vector of n_t observations $z_t = \{z_{t,i} : i = 1, 2, \cdots, n_t\}$. Assume that if these observations were acquired without any noise, then there would be a geometric relationship $m_l = h(x_t, z_{t,i})$, provided the i-th observation at the time t corresponds to the label l. These landmark-observation matchings are contemplated in a vector $c_t = \{c_{t,i} : i = 1, 2, \cdots, n_t\}$, where $c_{t,i} = l$ if $z_{t,i}$ (i-th observation at time t) corresponds to the label l. However, the observations are always noisy, and therefore it can only be expected that $h(x_t, z_{t,i})$ be a geometric estimation of $m_{c_{t,i}}$.

Let $\boldsymbol{x}$, $\boldsymbol{m}$, $\boldsymbol{c}$, $\boldsymbol{z}$ and $\boldsymbol{u}$ be the vectors whose components are the variables x_t, m_l, c_t, z_t and u_t, respectively.

The SLAM problem consists of the autolocation of a robot while the environment in which it operates is being mapped. This is translated into maximizing $P(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{c}|\boldsymbol{z}, \boldsymbol{u})$, where P is the probability distribution which characterizes the model. Subsequently, the SLAM problem will be modeled in the MRF context, and it will be resolved via ICM.

3.1 SLAM in MRF context

In order to give a solution for the SLAM problem in a MRF context, the graph and the potentials that characterize it must be defined. Regarding the graph, it is considered a node for each $x_t \in X_t$, $t \geq 1$; for each $c_t \in C_t = \{1, \cdots, L\}^{n_t}$, $t \geq 0$; and a single node for the map $\boldsymbol{m} \in M$. The initial state x_0 is a known fixed input. Then, the set of all possible states is defined by $\Omega = M \times \prod_{t=1}^{T_f} \boldsymbol{X}_t \times \prod_{t=0}^{T_f} \boldsymbol{C}_t$. The observations z_t, the control actions u_t, and the odometry data x_t are also external inputs to the model, since they are constant during the optimization process.

The graph edges are contemplated according to the probabilistic dependence existing on the model. The graph selected in this paper to contextualize the SLAM problem in the MRF theory is diagramed in Fig. 2. This graph was built following two premises: that it contains the fewest amount possible of edges; and that it contains the cliques of the form $\{x_{t-1}, x_t\}$ $(t \geq 2)$, $\{x_t, \boldsymbol{m}, c_t\}$ $(t \geq 1)$, and the initial $\{\boldsymbol{m}, c_0\}$.

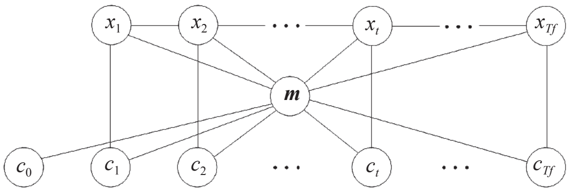


Fig. 2 SLAM graph in MRF context

These cliques allow to include a motion model and an observation model to the SLAM problem. In addition, the cliques $\{\boldsymbol{m}\}$ and $\{x_t\}$ $(t \geq 1)$ are considered to include a prior probabilistic model of the map and sensory fusion, respectively.

Although the methodology works independently of the chosen norm or distance, it is considered the Mahalanobis norm defined as $x_\Sigma = \sqrt{x^{\mathrm{T}}\Sigma x}$ for all $x \in \mathbf{R}^n$, with $\Sigma \in \mathbf{R}^{n\times n}$ as a positive definite matrix.

The potentials that include the motion model in the MRF model are defined by

$$\begin{aligned} &\Phi_{\{x_1\}}\left(x_1\right) = -\left\|g\left(x_0, u_0\right) - x_1\right\|_R^2 \\ &\Phi_{\{x_{t-1}, x_t\}}\left(x_{t-1}, x_t\right) = -\left\|g\left(x_{t-1}, u_{t-1}\right) - x_t\right\|_R^2, \quad t \geq 2 \end{aligned} \tag{3}$$

where $R > 0$ is a design matrix. These potentials link vehicle states of consecutive sampling periods, highlighting pairs of state vectors (x_{t-1}, x_t) that best match with the control actions and the motion model in ideal conditions. Note that kinematic and dynamic models can be used as the basis for their definitions.

Other sensory information that can be incorporated are the odometry data $\widehat{x}_t$, which are defined in this paper as an estimate of x_t based on the proprioceptive sensory information. In practice, when $t \gg 0$ is large, $\widehat{x}_t$ is not a good estimate of x_t, due to the cumulative odometry error. However, $x_t - x_{t-1}$ can be estimated by using $\widehat{x}_t - \widehat{x}_{t-1}$, which can be helpful in time intervals in which there are few useful observations that feed back the SLAM process. This information can be included through the following redefinition of the potentials (3)

$$\begin{aligned}
&\Phi_{\{x_1\}}\left(x_1\right) = -\left\|g\left(x_0, u_0\right) - x_1\right\|_R^2 - \left\|x_1 - \hat{x}_1\right\|_S^2 \\
&\Phi_{\{x_{t-1}, x_t\}}\left(x_{t-1}, x_t\right) = -\left\|g\left(x_{t-1}, u_{t-1}\right) - x_t\right\|_R^2 - \\
&\qquad \left\|x_t - x_{t-1} - \hat{x}_t + \hat{x}_{t-1}\right\|_S^2, \quad t \geq 2
\end{aligned} \tag{4}$$

where $S > 0$ is a design matrix.

On the other side, the potentials that characterize the observation model are

$$\begin{aligned}
&\Phi_{\{x_t, \boldsymbol{m}, c_t\}}(x_t, \boldsymbol{m}, c_t) = -\sum_{i=1}^{n_t} \left\|h\left(x_t, z_{t,i}\right) - m_{c_{t,i}}\right\|_Q^2, \ t \geq 1 \\
&\Phi_{\{\boldsymbol{m}, c_0\}}\left(\boldsymbol{m}, c_0\right) = -\sum_{i=1}^{n_0} \left\|h\left(x_0, z_{0,i}\right) - m_{c_{0,i}}\right\|_Q^2
\end{aligned}$$

where $Q > 0$ is a design matrix. These potentials prioritize configuration linkages $(x_t, \boldsymbol{m}, c_t)$ that best match with the observation model under ideal conditions.

The matrices R, Q and S allow granting greater or lesser influence to the potentials according to the confidence degree of the involved sensors.

Finally, in many applications, it is desirable to give greater weight to maps with certain characteristics. For example, it can be boosted maps in which the landmarks are at least at a predetermined distance, or maps without landmarks in certain regions. Such specifications can be included in a probabilistic model defined on the space of all possible environment maps. To this end, it is defined a potential function $\Phi_{\{\boldsymbol{m}\}}(\boldsymbol{m})$ that assigns, for example, a very large value to the desirable maps and 0 (a null value) to the undesirable maps. This way, any prior available information about the map can be included in the map model.

All these potentials are responsible for making prevail those states x_t, locations m_l, and matchings c_t, that best explain the observations z_t, that best suit to the control actions u_t, and that best fit to the odometry data. The potentials for the remaining cliques are defined equal to null potential, because it is not desired to use them. By definition, they will allow that R, Q or S to be null matrices. In this case, although $\|\cdot\|_R$, $\|\cdot\|_Q$ or $\|\cdot\|_S$ are not norms, the potentials are defined as nulls and their associated data are ignored by the model.

Then, the Gibbs probability associated to the SLAM problem is given by

$$\begin{aligned}
P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c} | \boldsymbol{z}, \boldsymbol{u}) \propto \exp\Bigg\{ &\Phi_{\{x_1\}}(x_1) + \\
&\sum_{t=2}^{T_f} \Phi_{\{x_{t-1}, x_t\}}(x_{t-1}, x_t) + \Phi_{\{\boldsymbol{m}, c_0\}}(\boldsymbol{m}, c_0) + \\
&\sum_{t=1}^{T_f} \Phi_{\{x_t, \boldsymbol{m}, c_t\}}(x_t, \boldsymbol{m}, c_t) + \Phi_{\{\boldsymbol{m}\}}(\boldsymbol{m}) \Bigg\}.
\end{aligned} \tag{5}$$

This modeling procedure allows to define potentials in many ways, providing great flexibility to exploit specific characteristics that are desired to be highlighted in the model.

In the paper, it is considered that $\boldsymbol{m}$ is a vector with the coordinates of the map landmarks. This definition is not the only possible since, for example, $\boldsymbol{m}$ can be an arrangement that assigns to each possible coordinate of the map a probability of locating a landmark. This infinite dimensional arrangement could be characterized, for example, by the means and covariances of a Gaussian mixture model, whose peaks represent the locations with the highest probability of observing a landmark. Alternatively, $\boldsymbol{m}$ can contain centers, lengths and orientations of a segment set. This can be useful for indoor applications.

This methodology allows also to include sensory fusion in a very natural way. In addition to the odometry data, the available GPS information can be included through the following potentials:

$$\Phi_{\{x_t\}}(x_t) = -1_T(t) \left\|\rho(x_t) - \breve{x}_t\right\|_K^2$$

$$\Phi_{\{x_t\}}(x_t) = \begin{cases} -\left\|\rho(x_t) - \breve{x}_t\right\|_K^2, & t \in T \\ 0, \quad t \notin T \end{cases}$$

where $K > 0$ is a design matrix, $T \subseteq \{1, 2, \cdots, T_f\}$ is the time subset in which the GPS data $\breve{x}_t$ is available, 1_T is the indicator function of T, and $\rho(x_t)$ is the robot position, which is supposed to be a subarray of x_t. However, the GPS information is not considered here in order to highlight the ICM ability to solve SLAM problems regardless of the GPS data availability.

This modeling methodology also admits the incorporation of dynamic objects. To this end, the variation of $\boldsymbol{m}$ over time should be allowed by firstly creating a node $\boldsymbol{m}_t$ for each t, and secondly by creating the nodes corresponding to $\boldsymbol{m}_{t-1}$ and $\boldsymbol{m}_t$. This linkage allows the locations given in $\boldsymbol{m}_{t-1}$ to be similar to those given in $\boldsymbol{m}_t$ according to the objects motion model, smoothing this way the errors introduced by bad measurements. Furthermore, a method should be incorporated that estimates the velocity vector $v_t = \{v_{t,l} : l = 1, \cdots, L\}$ of the L objects for any time t, and defines new potentials given by

$$\Phi_{\{\boldsymbol{m}_{t-1}, \boldsymbol{m}_t\}}(\boldsymbol{m}_{t-1}, \boldsymbol{m}_t) = -\sum_{l=1}^{L} \left\|\boldsymbol{m}_{t,l} - \gamma\left(\boldsymbol{m}_{t-1,l}, \boldsymbol{v}_{t-1,l}\right)\right\|_2^2$$

where γ is a function that kinematically characterizes the object motion in the absence of disturbances.

The multi robot SLAM problem can also be modeled by MRF. To this end, additional states $x_t^{(i)}$, $i = 1, \cdots, n$, should be generated by creating links between the states $x_t^{(i)}$ and $x_t^{(j)}$ when one of the robots observe other robot at time t. These links lead to the modeling of cooperative multi-robot SLAM problem.

3.2 ICM in SLAM problem

In this Section, the ICM iterative procedure is particularized to solve the SLAM problem defined in Section 3.1. From (2), the following conditional probabilities are calculated:

$$P(x_t|\boldsymbol{x}_{-t}, \boldsymbol{m}, \boldsymbol{c}, \boldsymbol{z}, \boldsymbol{u}) = P(x_t|x_{t-1}, x_{t+1}, \boldsymbol{m}, c_t, z_t, u_t) \propto \\ \exp\Bigg\{ - \|g(x_{t-1}, u_{t-1}) - x_t\|_R^2 - \|g(x_t, u_t) - x_{t+1}\|_R^2 - \\ \|x_t - x_{t-1} - \hat{x}_t + \hat{x}_{t-1}\|_S^2 - \|x_{t+1} - x_t - \hat{x}_{t+1} + \hat{x}_t\|_S^2 - \\ \sum_{i=1}^{n_t} \left\|h\left(x_t, z_{t,i}\right) - m_{c_{t,i}}\right\|_Q^2 \Bigg\} \tag{6}$$

$$P(c_t|\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{c}_{-t}, \boldsymbol{z}, \boldsymbol{u}) = \prod_{i=1}^{n_t} P(z_{t,i}|x_t, m_{c_{t,i}}, c_{t,i}) \propto \\ \exp\left\{ - \sum_{i=1}^{n_t} \left\|h\left(x_t, z_{t,i}\right) - m_{c_{t,i}}\right\|_Q^2 \right\} \tag{7}$$

$$P(\boldsymbol{m}|\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{z}, \boldsymbol{u}) \propto \\ \exp\left\{ \Phi_{\{\boldsymbol{m}\}}(\boldsymbol{m}) - \sum_{t=0}^{T_f} \sum_{i=1}^{n_t} \left\|h\left(x_t, z_{t,i}\right) - m_{c_{t,i}}\right\|_Q^2 \right\}. \tag{8}$$

The solution to the SLAM problem by using the iterative method ICM that is proposed in this paper is summarized in Algorithm 2. Superscripts are used in the notation in order to distinguish the states of different iterations. The Algorithm 2 consists of two steps: initialization (iteration 0) and ICM iterations.

In the initialization step, an initial state configuration for all the nodes is chosen. The initial states $x_t^{(0)}$, $\boldsymbol{m}^{(0)}$ and $c_t^{(0)}$ can be selected randomly, although the sensitivity of ICM to the initial point requires a more appropriate choice. One option is to choose the maximum of the conditional probabilities (6) to (8) in a time order ignoring the terms that require information at $t + 1$. This allows to include an on-line initialization in such a way that the process resembles an ICM iteration as close as possible.

Having a prior estimation of the map provides the ideal opportunity to choose $\boldsymbol{m}^{(0)} = \{m_l^{(0)} : l = 1, \cdots, L\}$. If this is not the case, each $m_l^{(0)}$ must be selected more finely. The locations $m_l^{(0)}$ are initialized when the l-th landmark is first seen with the mean of the estimated positions. The numbering of the landmarks is performed according to the appearance order.

Note that the initialization of the matchings $c_{t,i}^{(0)}$ requires the knowledge of $x_t^{(0)}$, as well as the initialization of $x_t^{(0)}$ requires the knowledge of these same matchings $c_{t,i}^{(0)}$. At this point, only x_0 is known. Hence, each $c_{0,i}^{(0)}$ is initialized with the label l if the location m_l is the closest to the estimated location $h(x_0, z_{0,i})$ (see (9)). Analogously, $c_{1,i}^{(0)}$ cannot be initialized since $x_1^{(0)}$ is still unknown. One possible choice is to consider $g(x_0, u_0)$ instead of $x_1^{(0)}$ (see (10) for $t = 1$). Finally, an alternative to initialize $x_1^{(0)}$ according to (6) but without $x_2^{(0)}$ is by using (11) with $t = 1$. All variables $c_{t,i}^{(0)}$ and $x_t^{(0)}$ are initialized by repeating this recursive procedure at time instant t (see (10) and (11)).

Note that certain required minimizations can be explicitly computed by setting to zero the partial derivatives (if these exist) of g and h.

Then, ICM iterations are performed in the step 2) of Algorithm 1. In this step, each node is visited according to a visit scheme, and the corresponding states are updated maximizing the respective conditional probabilities (6)–(8). This procedure has a great similarity with the recursive procedure of variable initialization (see (12)–(15)). The number N of ICM iterations is as large as necessary, and depends on the quality of the initialization.

As it will be shown in Sections 5 and 6, Algorithm 2 has a very good performance although the starting point moderately differs from the configuration that optimizes the posterior probability of the SLAM model. Algorithm 2 is by definition an algorithm to be run off-line, because data at time $t + 1$ are needed to update states of the time t in its step 2).

However, in order to run the algorithm on-line, it is proposed to run on-line the initialization step 1), and then perform the ICM updates at key moments, which can be among others: at the end; and in each closed loop, i.e., each time the robot revisits a key area and looks again the landmarks that it had left behind.

Algorithm 2. ICM-SLAM

1) Initialization: (on-line)

a) Each location $m_l^{(0)}$ is initialized when the l-th landmark is first seen.

b) Calculate $c_{0,i}^{(0)}$, $i = 1, \cdots, n_0$, with

$$c_{0,i}^{(0)} = \arg\min_l \left\|h\left(x_0, z_{0,i}\right) - m_l^{(0)}\right\|_Q^2 \tag{9}$$

c) for $t = 1 : T_f$

i) Calculate $c_{t,i}^{(0)}, i = 1, \cdots, n_1$

$$c_{t,i}^{(0)} = \arg\min_l \left\|h(g(x_{t-1}, u_{t-1}), z_{t,i}) - m_l^{(0)}\right\|_Q^2 \tag{10}$$

ii) Calculate $x_t^{(0)}$ with

$$x_t^{(0)} = \arg\min_x \Big\{ \|g(x_{t-1}, u_{t-1}) - x\|_R^2 + \|x - \hat{x}_t\|_S^2 + \sum_{i=1}^{n_t} \Big\| h\left(x, z_{t,i}\right) - m^{(0)}_{c^{(0)}_{t,i}} \Big\|_Q^2 \Big\} \tag{11}$$

end for

2) ICM iterations: (at key moments)

a) The visit scheme chosen is: c_0, c_1, x_1, c_2, x_2, $\cdots$, c_{T_f}, x_{T_f} and $\boldsymbol{m}$.

b) for $n = 1 : N$

i) Calculate $c_{0,i}^{(n)}, i = 1, \cdots, n_t$, with

$$c_{0,i}^{(n)} = \arg\min_l \left\| h\left(x_0, z_{0,i}\right) - m_l^{(n-1)} \right\|_Q^2 \tag{12}$$

ii) for $t = 1 : T_f$

Calculate $c_{t,i}^{(n)}, i = 1, \cdots, n_t$, with

$$c_{t,i}^{(n)} = \arg\min_l \left\| h(x_t^{(n-1)}, z_{t,i}) - m_l^{(n-1)} \right\|_Q^2 \tag{13}$$

Calculate $x_t^{(n)}$ with

$$\begin{aligned} x_t^{(n)} = \arg\min_x \Big\{ & \left\| g(x_{t-1}^{(n)}, u_{t-1}) - x \right\|_R^2 + \\ & \left\| g(x, u_t) - x_{t+1}^{(n-1)} \right\|_R^2 + \\ & \left\| x - x_{t-1}^{(n)} - \hat{x}_t + \hat{x}_{t-1} \right\|_S^2 + \\ & \left\| x_{t+1}^{(n-1)} - x - \hat{x}_{t+1} + \hat{x}_t \right\|_S^2 + \\ & \sum_{i=1}^{n_t} h(x, z_{t,i}) - m_{c_{t,i}^{(n)}}^{(n-1)}{}_Q^2 \Big\} \end{aligned} \tag{14}$$

end for

iii) Calculate $\boldsymbol{m}^{(n)}$ with

$$\boldsymbol{m}^{(n)} = \arg\min_{\boldsymbol{m}} \Big\{ -\Phi_{\{\boldsymbol{m}\}}(y) + \sum_{t=0}^{T_f} \sum_{i=1}^{n_t} \Big\| h\left(x_t^{(n)}, z_{t,i}\right) - m_{c_{t,i}^{(n)}} \Big\|_Q^2 \Big\} \tag{15}$$

end for

4 Example of SLAM problem

This section presents the specific SLAM problem that will be used to simulate and experience Algorithm 2. Consider a differential drive mobile robot (DDMR) as shown in Fig. 3. The pose of the DDMR is characterized by the state vector $x_t = [x_{t,x}, x_{t,y}, x_{t,\theta}]^{\mathrm{T}}$, where $(x_{t,x}, x_{t,y}) \in \mathbf{R}^2$ is its location and $x_{t,\theta} \in [0, 2\pi)$ is its orientation at time t. Consider also a kinematic model for the DDMR, in which the control action is $u_t = [\nu_t, \omega_t]$, where ν_t and ω_t are the linear and angular reference velocities.

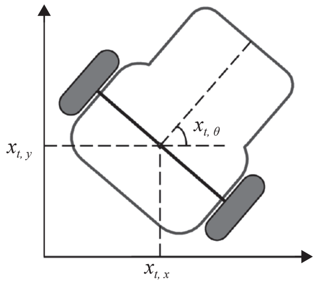


Fig. 3 Diagram of a DDMR

Then, the function that characterizes the robot motion model is given by

$$g(x_{t-1}, u_{t-1}) = x_{t-1} + \Delta t \begin{bmatrix} \nu_{t-1}\cos(x_{t-1,\theta}) \\ \nu_{t-1}\sin(x_{t-1,\theta}) \\ \omega_{t-1} \end{bmatrix}$$

where $\Delta t > 0$ is a fixed sampling period.

It is assumed that in the environment, there are L landmarks located at $m_l \in \mathbf{R}^2$ as indicated in Fig. 4.

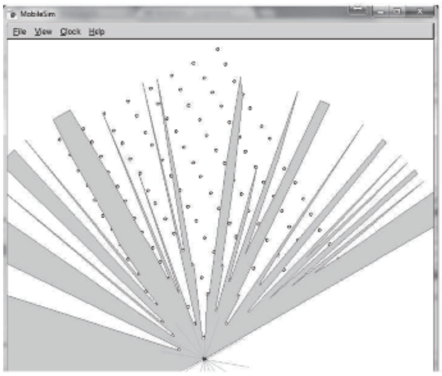


Fig. 4 SLAM simulation setting

The DDMR is equipped with a laser range finder that estimates the distance and direction to the landmarks located in the observation range (see Fig. 4). Suppose that at time t, this sensory information can be synthesized through $z_t = \{z_{t,i} : i = 1, \cdots, n_t\}$, where each $z_{t,i} = [z_{t,i,d}, z_{t,i,\theta}]$ contains the distance $z_{t,i,d}$ from the robot to the landmark $c_{t,i}$, and the direction $z_{t,i,\theta}$ from which the landmark was observed. This direction takes values between 0 (right side of the robot) and π (left side of the robot). Then, the function that characterizes the observation model is

$$h(x_t, z_{t,i}) = \begin{bmatrix} x_{t,x} \\ x_{t,y} \end{bmatrix} + z_{t,i,d} \begin{bmatrix} \cos\left(z_{t,i,\theta} + x_{t,\theta} - \dfrac{\pi}{2}\right) \\ \sin\left(z_{t,i,\theta} + x_{t,\theta} - \dfrac{\pi}{2}\right) \end{bmatrix}.$$

In addition, DDMR estimates its pose through encoders. In this context, the odometry information can be introduced into the potentials (4) more efficiently. Note that $\widehat{x}_t - \widehat{x}_{t-1}$ is a good estimator of $x_t - x_{t-1}$ if $\widehat{x}_{t-1,\theta} = x_{t-1,\theta}$, but if this is not the case, it is needed to properly rotate these vectors in order to obtain better approximations. To this aim, let $\tau(\theta)$ be the rotation matrix given by

$$\tau(\theta) = \left[\begin{array}{ccc} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{array}\right].$$

Then, $\tau(\widehat{x}_{t-1,\theta})(\widehat{x}_t - \widehat{x}_{t-1})$ is a good estimator of $\tau(x_{t-1,\theta})(x_t - x_{t-1})$, and the potentials (4) can be redefined by

$$\begin{aligned}\Phi_{\{x_{t-1},x_t\}}(x_{t-1},x_t) = &-\|g(x_{t-1},u_{t-1}) - x_t\|_R^2 - \\ &\|\tau(x_{t-1,\theta})(x_t - x_{t-1}) - \tau(\hat{x}_{t-1,\theta})(\hat{x}_t - \hat{x}_{t-1})\|_S^2, \\ &t \geq 2.\end{aligned} \quad (16)$$

Different labels are occasionally assigned to observations of the same landmark. The locations of these labels tend to be similar when the number of observations increases. In order to solve this problem and to formalize this fusion, it is defined a potential function $\Phi_{\{\boldsymbol{m}\}}(\boldsymbol{m}) = 0$ for the maps that include at least two landmarks inside a region defined by a certain threshold $\vartheta > 0$. Additionally, $\Phi_{\{\boldsymbol{m}\}}(\boldsymbol{m}) \gg 0$ values are assigned to the other maps, i.e., the potential of feasible maps is constant and large enough allowing to always choose some of these maps as optimal.

The particularization of Algorithm 2 begins with the selection of an appropriate method to perform the computations of the discrete optimizations (9), (10), (12) and (13) whereas minimizations (11) and (14) are performed by numerical methods (see Section 5 for more details). Equation (15) is solved by considering a three step procedure. First, the map locations are updated as follows:

$$m_l^{(n)} = \frac{1}{\sum\limits_{t=1}^{T_f} \#\{i : c_{t,i}^{(n)} = l\}} \sum_{t=1}^{T_f} \sum_{i:c_{t,i}^{(n)}=l} h(x_t^{(n)}, z_{t,i}). \quad (17)$$

Second, the label set is partitioned according to the distances between their locations by following two rules:

1) Two labels l_1 and l_2 must belong to the same component of the partition if $\|m_{l_1}^{(n)} - m_{l_2}^{(n)}\| < \vartheta$.

2) A label l_1 cannot belong to a component in which $\|m_{l_1}^{(n)} - m_{l_2}^{(n)}\| \geq \vartheta$ for all labels l_2 of the component.

Third, the matchings are redefined by grouping the observations with labels in the same component, and the locations are recalculated by using (17) again.

This procedure is also performed to finish the step 1) of Algorithm 2. Notice that this procedure assigns labels to objects seen only once. In practice, it is impossible to ignore the presence of sensing errors due to the appearance of many kinds of external objects. These errors produce false landmarks in the map that should be filtered. One way to do this is deleting the objects that were observed an amount of time less than a certain threshold $\zeta > 0$. This threshold should be chosen well below than the mean of the number of times that each landmark is expected to be seen during experimentation. This can easily be done by defining a counter through which the map locations m_l are removed if $\sum_{t=1}^{T_f} \sum_{i=1}^{n_t} \delta_l(c_{t,i}) < \zeta$, with $\delta_l(c_{t,i}) = 1$ if $c_{t,i} = l$, and $\delta_l(c_{t,i}) = 0$ in other case. This post-filtering can be applied each time $\boldsymbol{m}$ is updated.

5 Simulations

This section presents simulation results for the SLAM problem described in Section 4. In order to simulate the proposal, Matlab is used for calculations, and MobileSim program is used to simulate the dynamic model of a robot Pioneer p3dx-sh (http://robots.mobilerobots.com/wiki/MobileSim). For the connection between Matlab and MobileSim, a C++ interface based on shared memory resource is used, which allows to read the laser range sensor and generate on-line control commands. The PC used to perform the simulation has a Processor Intel@Core$^{\text{TM}}$ i7-4790 3.60 GHz and 16 GB of RAM. The numerical resolution of (11) and (14) is performed by using the Matlab function *fminunc*, with initial point setting by the kinematic model and default options. The optimizations (9), (10), (12) and (13) are solved by using the Matlab optimization functions named *cluster, linkage* and *pdist2.*

The initial pose of DDMR and the landmarks considered are shown in Fig. 4. A control method to allow the robot to follow an irregular path was implemented. While the robot navigated, it acquired laser observations of distances and directions (with respect to robot) of the landmark in the operation radius. In order to reduce the stored information, maximum range observations were eliminated, since the achievement of this value indicates that there are not landmarks in these directions. In addition, isolated observations were eliminated, i.e., those observations that were not accompanied by a similar observation in the contiguous directions. This was done based on the fact that any observed landmark to a logical distance produces at least two measurements in adjacent orientations, and that isolated observations generally come from sensing errors. Adjacent observations are introduced to the algorithm as different observations. Regarding the parameters of the model, it is considered that:

1) R and Q are identity matrices to give equal weight

to the motion and observation models,

2) S is the null matrix because the odometry error in MobileSim is almost negligible over short distances and it makes no sense to show the ICM virtues using these data.

3) $\vartheta = 1$ meter and $\zeta = 500$ observations.

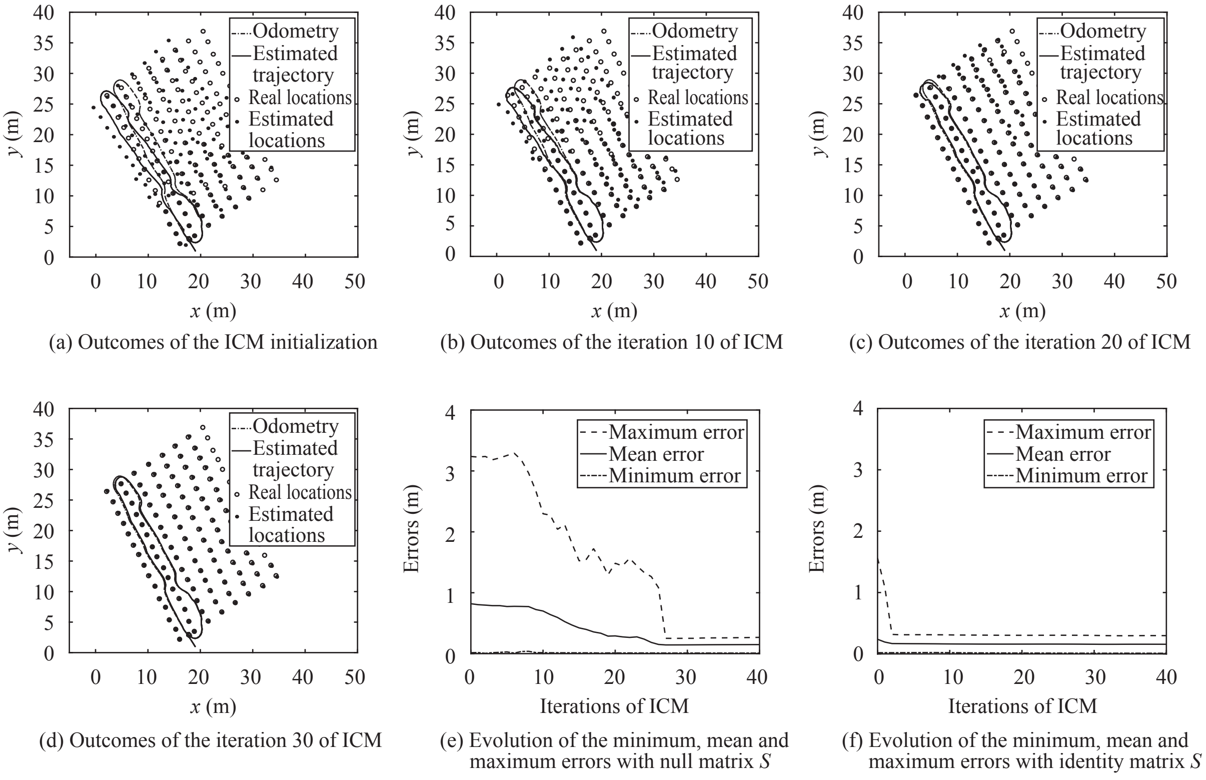
Note that 500 is small since there are multiple observations for each landmark at each sampling period of $\Delta t = 0.1$.

Results of the first simulation are summarized in Fig. 5. The circles show the landmarks real locations, and the path reconstructed by using odometric data is plotted in dash-dot line. In Fig. 5(a), the estimated trajectory appears in solid line whereas the points show the estimated locations of the landmarks at the end of step 1) of Algorithm 2. These are the initial estimates of ICM, and based on them, the ICM iterations are produced. It can be observed that the estimated trajectory is not perfectly followed with estimated landmarks on the sides of such path. In Figs. 5(b), 5(c) and 5(d), the outputs of the ICM iterations 10, 20 and 30 are shown. Note how iteration by iteration ICM corrects the deviation of the initial estimate of the path until it converges to a trajectory very similar to the actual one. At the same time, it is shown how the algorithm corrects the locations until it reaches very good estimates of the sensed landmarks.

In order to calculate the errors, it is measured the distance between each estimated landmark and its nearest real location. The minimum, maximum and mean errors with respect to the observed landmark are plotted in Fig. 6(c). The final mean error is 0.148 m. The estimated errors in the first iterations are underestimations, because there are very bad estimates of a landmark A that coincides with the real location of another landmark B. This estimation has null error in the error computation, but in fact the error is much greater. However, this error calculation methodology is the simplest and most general, allowing to measure errors in extreme situations as shown in Fig. 6(a) and to evidence the ICM convergence.

Another peculiarity to consider is that the maximum errors are observed in estimations of landmark locations that are not in the adjacencies of the robot paths. If the mean error is calculated based solely on the estimated position of the landmarks of the three adjacent rows to the robot trajectory, then the mean error is 0.112 m. It can be seen that in the iteration 27, the algorithm already reached the convergence, requiring a total time of 432.24 s. Most of the computational cost is consumed in the iterative resolution of (11) and (14). Note that ICM converges much faster if odometry data are used (see Fig. 5(f)).

One of the main sources of error comes from using the maximum amplitude of the laser range, producing the initial estimates shown in Fig. 5(a). It is observed that reducing this range produces better initial maps, and therefore fewer ICM iterations to correct initial mistakes are required. Fig. 7(a) shows the ICM initialization output for a maximum laser value of 10 m; whereas the outputs convergence for the ICM iteration 10 appears in Fig. 7(b). It



(a) Outcomes of the ICM initialization

(b) Outcomes of the iteration 10 of ICM

(c) Outcomes of the iteration 20 of ICM

(d) Outcomes of the iteration 30 of ICM

(e) Evolution of the minimum, mean and maximum errors with null matrix S

(f) Evolution of the minimum, mean and maximum errors with identity matrix S

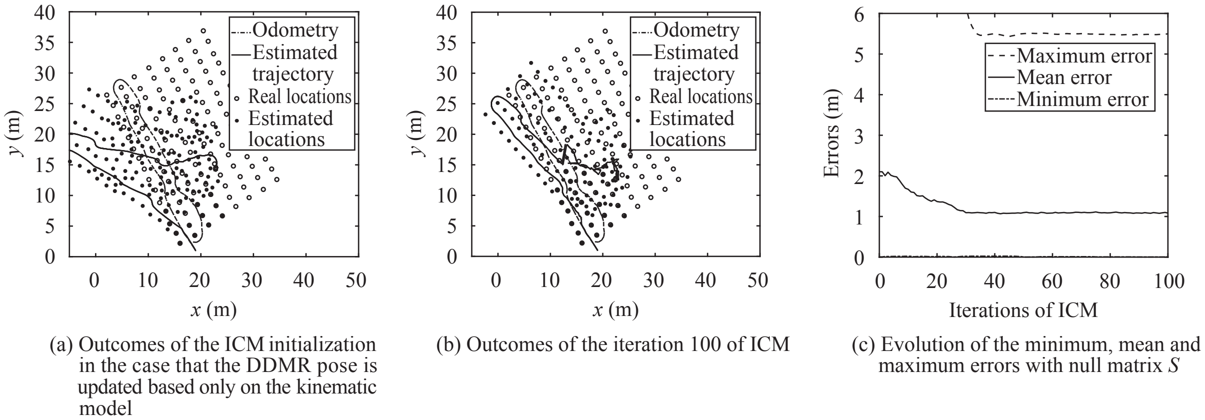
Fig. 5 Outcomes of Simulation 1

is noted that ICM converges close to its initialization, producing little change in subsequent iterations. In Fig. 7(d), the error evolution when considering the odometry data through the identity matrix S is shown. Also in this case, the inclusion of odometric data accelerates the algorithm convergence.

In both simulations, ICM converges to a local maximum of the posterior probability of Gibbs that characterizes the SLAM problem. This local maximum is the closest to the set of initial estimates, and therefore requires good initial estimates to optimize the ICM performance. However, ICM works well despite being initialized with the estimates showed in Simulation 1. Fig. 6 reports the outcomes when the ICM is initialized with the kinematic estimates (see Fig. 6(a), i.e., instead of using a numerical optimizer to solve (11) and its equivalent at time t, the vehicle poses are initialized by using its kinematic model only. With respect to the ICM sensitivity to very bad initializations, it can be observed the important role of the vehicle pose numerical updates at each ICM iteration.

6 Experiment

An experiment was carried out in order to verify the performance of the proposed algorithm. A Pioneer 3At robot from ActiveMedia equipped with an outdoor laser range-finder sensor from Sick (see Fig. 8(a)) was teleoper-

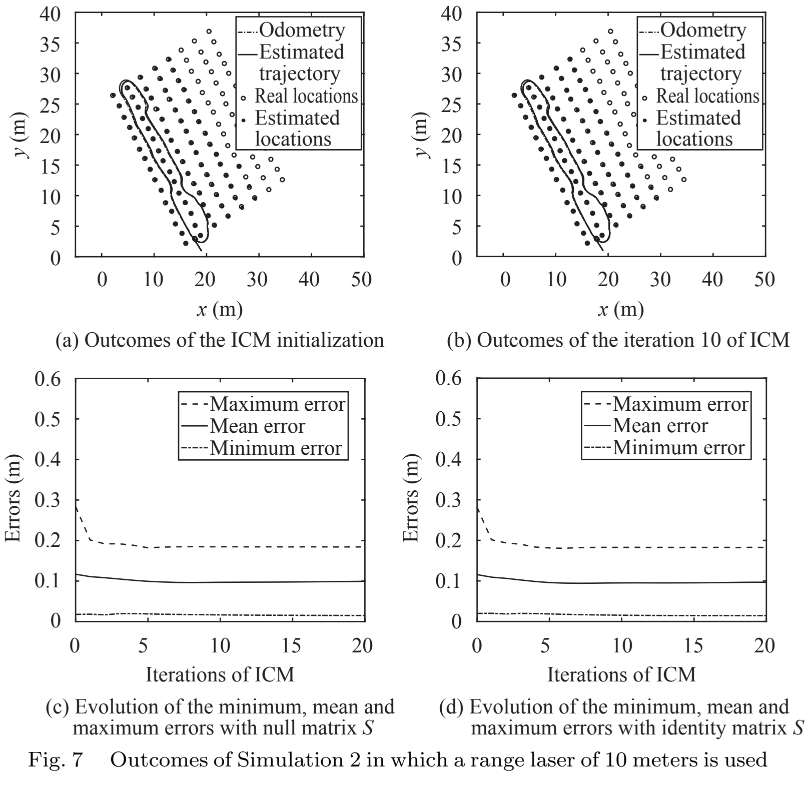


(a) Outcomes of the ICM initialization in the case that the DDMR pose is updated based only on the kinematic model

(b) Outcomes of the iteration 100 of ICM

(c) Evolution of the minimum, mean and maximum errors with null matrix S

Fig. 6 Outcomes of ICM if it is initialized with very bad estimations of the states



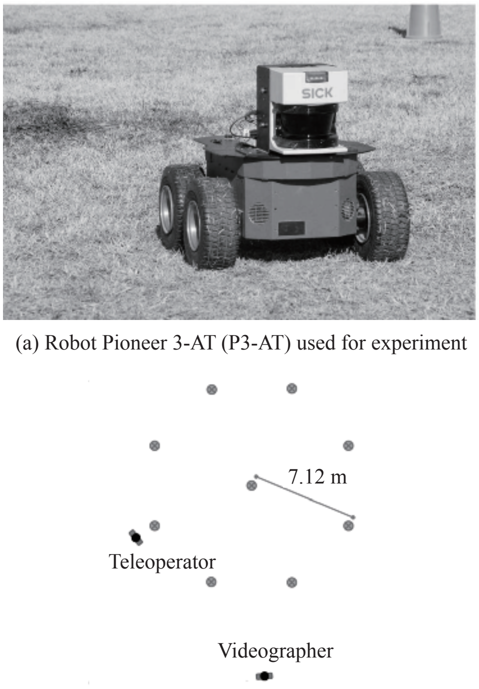
(a) Outcomes of the ICM initialization

(b) Outcomes of the iteration 10 of ICM

(c) Evolution of the minimum, mean and maximum errors with null matrix S

(d) Evolution of the minimum, mean and maximum errors with identity matrix S

Fig. 7 Outcomes of Simulation 2 in which a range laser of 10 meters is used



(a) Robot Pioneer 3-AT (P3-AT) used for experiment

(b) Diagram of the environment

Fig. 8 Experimental setting

ated in an open field setting. The environment consists of 11 static reference points: a teleoperator, a videographer, 8 cones that form a regular octagon with a radius of 7.12 m, and a cone on the center of the octagon (see Fig. 8(b)).

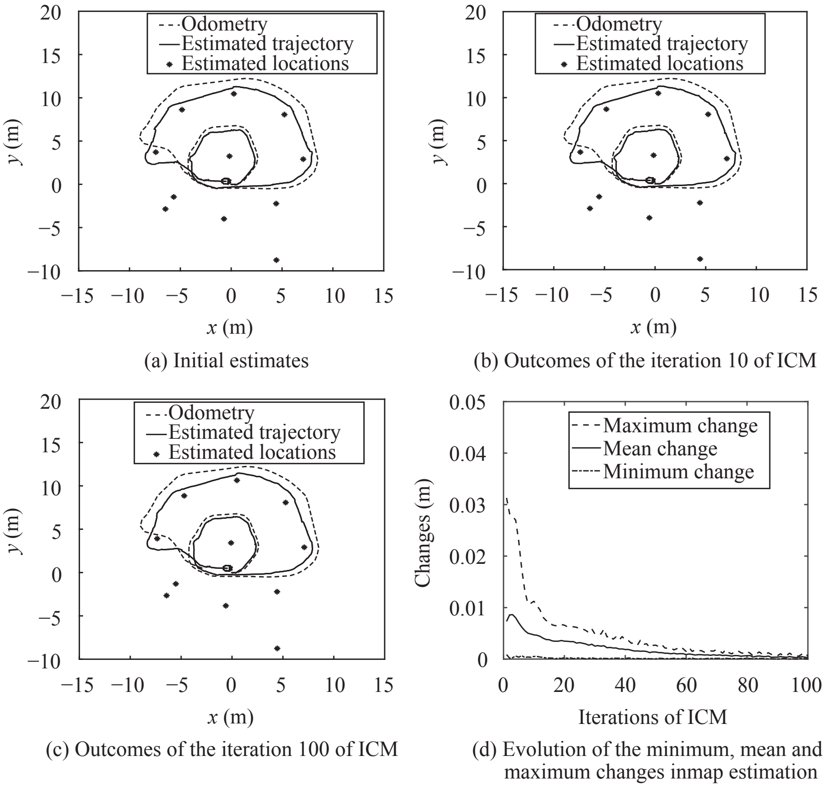
The parameters fixed for the experiment are:

1) R, Q and S are identity matrices to give equal weight to the motion model, observation models, and the odometry data.

2) $\vartheta = 1$ meter and $\zeta = 500$ observations.

The experimentation results are shown in Fig. 9. The odometry path is shown with dashed lines, the trajectory estimations are shown with solid lines, and the estimated location of the landmarks are shown with asterisks markers. In Fig. 9(a), the initial ICM estimates are shown, and in Figs. 9(b) and 9(c), the ICM estimates in its iterations 10 and 100 are displayed respectively. In Fig. 9(d), the evolution of the average norm of the changes in the map estimations from one iteration to another is shown. Note that the estimates practically converge in the first iterations.

The computation time of the initial ICM estimations is about 10.8 s for the complete dataset (see Fig. 9(a)). Taking into account that the data acquisition lasted 183 s with a sample time of 0.1 s, this gives an average time of approximately 6 milliseconds per sample time dedicated to obtain the initial estimation. This shows that the algorithm can be initialized on-line. On the other hand, each ICM iteration also required approximately 10.8 s. As shown in Fig. 9(d), the maximum change from the initial estimation is less than 35 millimeters. This clearly exposes the good performance of the initial estimation. Ad-



(a) Initial estimates

(b) Outcomes of the iteration 10 of ICM

(c) Outcomes of the iteration 100 of ICM

(d) Evolution of the minimum, mean and maximum changes inmap estimation

Fig. 9 Experiment outcomes

ditionally, few ICM iterations are required in order to achieve that errors converge. This shows that initial errors can be quickly corrected through ICM iterations.

7 Conclusions

In this paper, the complex SLAM problem is modeled through a very flexible MRF, which can be used as a global framework for particular applications. Among the potentialities of the methodology, it can be mentioned: It allows to incorporate an observation model regardless of the type of sensor being chosen. It allows to incorporate any motion model. It allows to incorporate prior information of the map through a map model. It allows to work with maps of diverse natures. It allows to incorporate sensor fusion weighted according to the accuracy thereof. It allows to incorporate multiple robots. It allows to work with dynamic environments.

Later, it is proposed to use the ICM algorithm to solve the SLAM problem through a probabilistic optimization. Among the characteristics of ICM, it can be mentioned: It has theoretical convergence regardless of the MRF chosen to model the SLAM problem and the nature of the multiple observations and measurements considered. It can be used as a post-processing methodology if it is initialized with estimates obtained with another SLAM solver. It can be initialized on-line and be iterated in parallel to correct the initial estimates whenever deemed appropriate. It is robust to initialization of intermediate quality.

Simulations and experiments show the flexibility and the excellent results of the proposal.

References

[1] H. Durrant-Whyte, T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006. DOI: 10.1109/MRA.2006.1638022.

[2] T. Bailey, H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006. DOI: 10.1109/MR.2006.1678144.

[3] T. S. Ho, Y. C. Fai, E. S. L. Ming. Simultaneous localization and mapping survey based on filtering techniques. In *Proceedings of the 10th Asian Control Conference*, IEEE, Kota Kinabalu, Malaysia, pp. 1–6, 2015.

[4] X. C. Zhang, B. Li, S. L. Joseph, J. Z. Xiao, Y. Sun, Y. L. Tian, J. P. Muñoz, C. C. Yi. A SLAM based semantic indoor navigation system for visually impaired users. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, Hong Kong, China, pp. 1458–1463, 2015.

[5] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, V. Lepetit. Instant outdoor localization and SLAM initialization from 2.5D maps. *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1309–1318, 2015. DOI: 10.1109/TVCG.2015.2459772.

[6] J. Kim. Aerial inertial-slam: Progresses and future challenges. In *Proceedings of the 12th International Conference on Ubiquitous Robots and Ambient Intelligence*, IEEE, Goyang, South Korea, pp. 212–213, 2015.

[7] L. Paull, S. Saeedi, M. Seto, H. Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014. DOI: 10.1109/JOE.2013.2278891.

[8] F. Hidalgo, T. Bräunl. Review of underwater SLAM techniques. In *Proceedings of the 6th International Conference on Automation, Robotics and Applications*, IEEE, Queenstown, New Zealand, pp. 306–311, 2015.

[9] C. L. Wang, T. M. Wang, J. H. Liang, Y. C. Zhang, Y. Zhou. Bearing-only visual SLAM for small unmanned aerial vehicles in GPS-denied environments. *International Journal of Automation and Computing*, vol. 10, no. 5, pp. 387–396, 2013. DOI: 10.1007/s11633-013-0735-8.

[10] D. Y. Bi, Y. L. Yu, Y. Shen, C. Melchiorri. Implementation of trapped personnel detection and evacuation guidance in indoor fire scene based on quadrotor UAV. In *Proceedings of International Conference on Intelligent Control and Computer Application*, Zhengzhou, China, pp. 57–62, 2016.

[11] S. Thrun, W. Burgard, D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, Cambridge, Mass, UK: The MIT Press, 2005.

[12] C. De la Cruz, T. Freire Bastos, F. A. Auat Cheein, R. Carelli. SLAM-based robotic wheelchair navigation system designed for confined spaces. In *Proceedings of IEEE International Symposium on Industrial Electronics*, IEEE, Taipei, China, pp. 2331–2336, 2010.

[13] F. Auat Cheein, G. Steiner, G. P. Paina, R. Carelli. Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. *Computers and Electronics in Agriculture*, vol. 78, no. 2, pp. 195–207, 2011. DOI: 10.1016/j.compag.2011.07.007.

[14] Y. Lee, J. Choi, H. T. Choi. Experimental results of real-time sonar-based underwater localization using landmarks. In *Proceedings of MTS/IEEE Washington OCEANS'15*, IEEE, Washington DC, USA, 2015.

[15] M. R. Nepali, N. Yadav, U. Dev, S. Mishra, S. L. Shrestha. A strategic methodology for 2D map building in an Indoor environment. In *Proceedings of the 1st International Conference on Next Generation Computing Technologies*, IEEE, Washington DC, USA, pp. 797–803, 2015.

[16] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J. M. Rendón-Mancha. Visual simultaneous localization and mapping: A survey. *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015. DOI: 10.1007/s10462-012-9365-8.

[17] S. C. Zhou, R. Yan, J. X. Li, Y. K. Chen, H. J. Tang. A brain-inspired SLAM system based on ORB features. *International Journal of Automation and Computing*, vol. 14, no. 5, pp. 564–575, 2017. DOI: 10.1007/s11633-017-1090-y.

[18] L. T. Hsu, Y. Wada, Y. L. Gu, S. Kamijo. Rectification of 3D building models based on GPS signal collected by vehicle. In *Proceedings of IEEE International Conference on Vehicular Electronics and Safety*, IEEE, Yokohama, Japan, pp. 132–139, 2015.

[19] C. S. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, C. Demonceaux. Static-map and dynamic object reconstruction in outdoor scenes using 3-D motion segmentation. *IEEE Robotics and Automation Letters*, vol. 1, no. 1,

pp. 324–331, 2016. DOI: 10.1109/LRA.2016.2517207.

[20] G. Bresson, R. Aufrére, R. Chapuis. A general consistent decentralized simultaneous localization and mapping solution. *Robotics and Autonomous Systems*, vol. 74, pp. 128–147, 2015. DOI: 10.1016/j.robot.2015.07.008.

[21] S. Saeedi, M. Trentini, M. Seto, H. Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016. DOI: 10.1002/rob.21620.

[22] S. Dominguez, B. Khomutenko, G. Garcia, P. Martinet. An optimization technique for positioning multiple maps for self-driving car's autonomous navigation. In *Proceedings of the 18th International Conference on Intelligent Transportation Systems*, IEEE, Las Palmas, Spain, pp. 2694–2699, 2015.

[23] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis. A quadratic-complexity observability-constrained unscented Kalman filter for SLAM. *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1226–1243, 2013. DOI: 10.1109/TRO.2013.2267991.

[24] J. A. Hesch, D. G. Kottas, S. L. Bowman, S. I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2014. DOI: 10.1109/TRO.2013.2277549.

[25] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993. DOI: 10.1007/BF01440847.

[26] M. Adams, B. N. Vo, R. Mahler, J. Mullane. SLAM gets a PHD: New concepts in map estimation. *IEEE Robotics & Automation Magazine*, vol. 21, no. 2, pp. 26–37, 2014.

[27] P. J. Huber. *Robust Statistics*, New York, USA: Wiley, 1981.

[28] R. Smith, M. Self, P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proceedings of the 4th International Symposium on Robotics Research*, Santa Clara, USA, pp. 467–474, 1987.

[29] P. Moutarlier, R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *Proceedings of the 5th International Symposium on Robotics Research*, Tokyo, Japan, pp. 85–94, 1989.

[30] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of AAAI National Conference on Artificial Intelligence*, Menlo Park, USA, pp. 593–598, 2002.

[31] T. Z. Lv, C. X. Zhao, H. F. Zhang. An improved Fast-SLAM algorithm based on revised genetic resampling and SR-UPF. *International Journal of Automation and Computing*, Online First. DOI: 10.1007/s11633-016-1050-y.

[32] M. A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, San Francisco, USA, pp. 1157–1164, 2003.

[33] F. Dellaert, M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006. DOI: 10.1177/0278364906072768.

[34] M. Jadaliha, J. Choi. Fully Bayesian simultaneous localization and spatial prediction using Gaussian Markov random fields (GMRFs). In *Proceedings of the American Control Conference*, IEEE, Washington DC, USA, pp. 4592–4597, 2013.

[35] Z. L. Xiao, H. K. Wen, A. Markham, N. Trigoni. Indoor tracking using undirected graphical models. *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2286–2301, 2015. DOI: 10.1109/TMC.2015.2398431.

[36] J. Nocedal, S. J. Wright, *Numerical Optimization*, Berlin, Germany: Springer Verlag, 2006.

[37] D. M. Rosen, C. DuHadway, J. J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Seattle, USA, pp. 5822–5829, 2015.

[38] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, vol. 6, no. 1–2, pp. 3–15, 1990. DOI: 10.1016/S0921-8890(05)80025-9.

[39] R. G. Cowell, P. A. Dawid, S. L. Lauritzen, D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*, New York, USA: Springer, 1999.

[40] J. Folkesson, H. Christensen. Graphical SLAM-a self-correcting map. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, New Orleans, USA, vol. 1, pp. 383–389, 2004.

[41] J. Sola. Course on SLAM, Technical Report, Robotics Institute and Industrial Computing, Polytechnic University of Catalonia, Barcelona, Spain, 2016.

[42] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, A. Y. Ng. Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results, Technical Report CMU-CS-02-112, Carnegie Mellon University, Pittsburgh, USA, 2002.

[43] G. Dissanayake, S. B. Williams, H. Durrant-Whyte, T. Bailey. Map management for efficient simultaneous localization and mapping (SLAM). *Autonomous Robots*, vol. 12, no. 3, pp. 267–286, 2002. DOI: 10.1023/A:1015217631658.

[44] Y. Latif, C. Cadena, J. Neira. Robust graph SLAM backends: A comparative analysis. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Chicago, USA, pp. 2683–2690, 2014.

[45] N. Carlevaris-Bianco, M. Kaess, R. M. Eustice. Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014. DOI: 10.1109/TRO.2014.2347571.

[46] B. Suger, G. D. Tipaldi, L. Spinello, W. Burgard. An approach to solving large-scale slam problems with a small memory footprint. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Hong Kong, China, pp. 3632–3637, 2014.

[47] M. Mazuran, G. D. Tipaldi, L. Spinello, W. Burgard. Nonlinear graph sparsification for SLAM. In *Proceedings of Robotics: Science and Systems*, Daejeon Convention Center, Berkeley, USA, 2014.

[48] S. Choudhary, V. Indelman, H. I. Christensen, F. Dellaert. Information-based reduced landmark SLAM. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Seattle, USA, pp. 4620–4627, 2015.

[49] R. Kindermann, J. L. Snell. *Markov Random Fields and Their Applications*, Providence, USA: American Mathematical Society, 1980.

[50] S. Z. Li. *Markov Random Field Modeling in Image Analysis*, 3rd ed., London, UK: Springer, 2009.

[51] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, *Series B*, vol. 48, no. 3, pp. 259–302, 1986.

[52] J. M. Hammersley, P. Clifford. Markov fields on finite graphs and lattices, Technical Report, University of California, Berkeley, USA, 1971.

[53] S. Geman, D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[54] A. Gelman, C. Robert, N. Chopin, J. Rousseau. *Bayesian Data Analysis*, London, UK: Chapman & Hall, 1995.

[55] S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi. Optimization by simulated annealing. *Science*, vol. 220, no. 4598, pp. 671–680, 1983. DOI: 10.1126/science.220.4598.671.

[56] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985. DOI: 10.1007/BF00940812.

J. Gimenez received the B. Sc. degree in mathematics from the National University of San Juan (UNSJ), Argentina in 2009, and the Ph. D. degree in mathematics from the National University of Córdoba (UNC), Argentina in 2014. Currently, he is an assistant researcher of the Argentinean National Council for Scientific Research (CONICET), and an adjunct professor in the Institute of Automatics, Argentina

His research interests include probabilistic and statistical implementations of robotics, such as simultaneous localization and mapping (SLAM) algorithms.

E-mail: jgimenez@inaut.unsj.edu.ar

ORCID iD: 0000-0002-1731-8520

A. Amicarelli received the B. Eng. degree in chemical engineering from the National University of San Juan, Argentina in 2000, and the Ph. D. degree in control systems from the Institute of Automatics (INAUT) of the same university in 2007. Her main works are related with state estimations of complex non-linear systems and with bioprocess control. She is an assistant researcher of the Argentinean National Council for Scientific Research from 2012.

Her research interests include systems modeling, process control and state estimation to control porpoises.

E-mail: amicarelli@inaut.unsj.edu.ar

ORCID iD: 0000-0002-4303-8776

J. M. Toibero received the B. Eng degree in electronic engineering from the Facultad Tecnológica Nacional of Argentina, Argentina in 2002, and the Ph. D. degree in control systems from the Institute of Automatics at the National University of San Juan, Argentina in 2007. His main works are related to nonlinear control of robotic platforms and robotics applications in agriculture. He is with the National Council for Scientific and Technological Research of Argentina since 2011, actually he is an adjunct researcher. He leads different technological projects and his current scientific research is at the Institute of Automatics of San Juan, Argentina.

His research interests include wheeled mobile robots, manipulators force/impedance, switched, hybrid, nonlinear control methods applied to automatic control and visual servoing.

E-mail: mtoibero@inaut.unsj.edu.ar (Corresponding author)

ORCID iD: 0000-0001-9347-566X

F. di Sciascio received the B. Sc. degree in electromechanical engineering with orientation in electronics from the National University of Buenos Aires (UBA), Argentina in 1986. He received the M. Sc. degree in engineering of control systems, and the Ph. D. degree in engineering, from the Institute of Automatics, National University of San Juan, Argentina in 1994 and 1997, respectively. Since 1987, he is a professor and researcher at the INAUT and he is currently a full professor in charge of the subjects artificial intelligence complements and identification and adaptive control in the Department of Electronics and Automation. At the same time, he is a professor of the postgraduate degree at the INAUT, Argentina.

His research interests include modelling, identification and estimation in dynamical systems, and technology developments in automatic process control.

E-mail: fernando@inaut.unsj.edu.ar

ORCID iD: 0000-0002-4382-6289

R. Carelli received the B. Sc. degree in engineering from the National University of San Juan, Argentina in 1976, and received the Ph. D. degree in electrical engineering from the National University of Mexico (UNAM), Mexico in 1989. He is a full professor at the National University of San Juan and a senior researcher of the National Council for Scientific and Technical Research, Argentina. He is director of the Automatics Institute, National University of San Juan, Argentina. He is a senior member of IEEE and a member of Argentine Association of Automatic Control - International Federation of Automatic Control (AADECA-IFAC).

His research interests include robotics, manufacturing systems, adaptive control and artificial intelligence applied to automatic control.

E-mail: rcarelli@inaut.unsj.edu.ar

ORCID iD: 0000-0003-0688-7020