MULTI-ROBOT SYSTEMS, TRENDS AND DEVELOPMENT

Edited by **Toshiyuki Yasuda and Kazuhiro Ohkura**

INTECHWEB.ORG

Multi-Robot Systems, Trends and Development
Edited by Toshiyuki Yasuda and Kazuhiro Ohkura

Published by InTech
Janeza Trdine 9, 51000 Rijeka, Croatia

Publishing Process Manager Ana Nikolic
Technical Editor Teodora Smiljanic
Cover Designer Martina Sirotic
Image Copyright Vladimir Nikitin, 2010. Used under license from Shutterstock.com

First published January, 2011
Printed in India

A free online edition of this book is available at www.intechopen.com
Additional hard copies can be obtained from orders@intechweb.org

Multi-Robot Systems, Trends and Development, Edited by Toshiyuki Yasuda and Kazuhiro Ohkura
p. cm.
ISBN 978-953-307-425-2

Contents

6

Building Visual Maps with a Team of Mobile Robots

Mónica Ballesta, Arturo Gil, Óscar Reinoso and Luis Payá
Miguel Hernández University
Elche

1. Introduction

This paper tackles the problem of Simultaneous Localization and Map Building (SLAM) carried out by a team of robots. Particularly these robots build landmark-based maps by extracting interest points from the environment. These points are characterized by a local descriptor and a 3D position on the environment, constituting the visual landmarks. In this approach we consider the situation in which the robots start their mapping task independently. That is to say, the path followed by the robots and the observations are estimated independently. After a while, there is a set of independent local maps that can be fused in order to obtain a global map. For that reason, we have also solved the problem of aligning and fusing visual maps. Finally, once a global map is obtained, the robots continue the map building task jointly. Regarding the sensors used in order to extract information from the environment, some authors use range sensors such as SONAR (Wijk & Christensen, 2000; Kwak et al., 2008) or LASER (Leonard & Durrant-Whyte, 1991; Thrun, 2001). However, in the last years, there is a great interest on using cameras as sensors. This approach is denoted as visual SLAM (Valls-Miró et al., 2006). The cameras are less expensive than laser and offer a higher amount of information from the environment. This advantage makes it possible to incorporate additional applications to the robot, such as face recognititon. Additionally, 3D information can be obtained from the environment when using stereo vision (Murray & Little, 2000; Gil, Reinoso, Fernández, Vicente, Rottmann & Martínez-Mozos, 2006). Most approaches using visual information are landmark-based. A process to extract these visual features with accuracy is required. In these sense, several detection and description methods appear in the literature such as the Harris Corner Detector (Davison & Murray, 2002), Harris-Laplace (Jensfelt et al., 2006), SIFT (Gil, Reinoso, Martínez-Mozos, Stachniss & Burgard, 2006; Little et al., 2002) and SURF (Murillo et al., 2007). In (Gil et al., 2009; Ballesta, Gil, Reinoso & Úbeda, 2010) we performed an evaluation comparing several detection and description methods in order to obtain the most suitable feature extractor for visual SLAM. As a result, we obtained that the Harris Corner Detector in combination with the u-SURF descriptor satisfied our requirements in visual SLAM. The task of building a map of the environment while simultaneously localizing in it can be performed by a single robot. However, it will be carried out with more efficiency if there is a team of robots which collaborates in this task. This approach is denoted as multi-robot SLAM (Konolige et al., 2003). In this field, two main solutions can be found. On the one hand, in some proposals the robots build a unique global map (Gil, Reinoso, Martínez-Mozos, Stachniss & Burgard, 2006; Fenwick et al., 2002). In this case, the exploration task can be performed more efficiently since the robots have a global notion of the environment. However, the initital pose of the robots must be known, which may not be possible in practice. On the other hand, some authors present the approach in which each robot builds its own map indepentently (Roumeliotis & Bekey, 2002; Stewart et al., 2003; Zhou & Roumeliotis, 2006). In this case, it is not necessary to know the initial positions of the robots. Nevertheless, a process of merging the data of the different robots is needed. In the situation that we set out, the robots start from different positions and build their maps independently. Therefore, the initial position of the robots is unknown. After a while, these independent local maps can be fused into a global map. Before fusing the local maps, the transformation that relates the different reference systems of the robots should be computed. This previous step is called map alignment. Then, once the relative positions of the robots is known and the maps can be expressed in the same reference sytem, the maps are fused into a global map. After solving the alignment problem, we propose that the robots continue their navigation tasks jointly. To sum up, the robots initially build their maps independently and then, once they manage to align and fuse their maps, the continue the map building together. In order to build the map, we propose a Rao-Blackwellized particle filter, firstly for a single robot and then extended to the multi-robot case. The paper is structured into three parts according to the approach proposed. First, in Section 2 we concentrate on the independent stage of the proposal, i.e., there is a team or robots that begins tis navigation tasks independently. In this case, we describe the particle filter used to solve the SLAM problem (Section 2.2). Logically, since the robots act indepdendently, the algorithm has been implemented for a single robot. Secondly, we concentrate in the fusion stage in Section 3. At this moment there is a set of local maps, independently built, that have to be aligned and merged into a global map. Finally, the third stage consist of the multi-robot map building (Section 4). After the fusion stage, the robots build together the map of the environment. In this case, the estimate of the map and the paths is performed jointly. For this purpose, we propose an extension of the previous particle filter to the multi-robot case. Finally, we present some experiments that validate the proposal (Section 5) and the main conclusions in Section 6.

2. Independent map building

In the first stage of the approach presented in this paper, there is a team of robots in which each one builds its map independently. The map building is performed by means of a Rao-Blackwellized particle filter, known with the general term FastSLAM (Montemerlo, 2003). The robots may start at different positions and they have no knowledge about their relative poses. After a while, the robots will have built several local maps that can be fused into a global map. This is done even if the relative initial positions of the robots are unknown. In addition, the robots compare their maps and compute the relative position between them, based on the descriptor simmilarity of the landmarks. In the following, we describe, in detail, the steps of the FastSLAM algorithm used by each robot independently. Then we focus on the map alignment step in which the relative position of the robots is obtained. Finally, it is explained how the local maps are fused into a global one.

2.1 Visual landmarks

In this paper we focus on a visual SLAM approach. Particularly, the robots observe distinctive points in the environment. These points are then located in a global reference system. In this case, the reference system of each robot is located in its initial position. The process of obtaining distinctive points from the environment is tackled by a feature extractor, i.e., the combination of a detector and a descriptor, that obtains suitable features. At the detection stage, it is desirable that the points can be detected from different viewpoints, due to the fact that the robots, while navigating, will observe the same from different position. As a result of a previous work (Gil et al., 2008), the Harris Corner detector (Harris & Stephens, 1998) was selected as the most suitable for this kind of maps, since it extracts stable points. Then, the description stage is also crucial for the extraction of good visual features. It is very important that the points are distinguishable enough, since this influences in the solution of the data association problem. When a robot performs an observation, it should be able to distinguish whether it is a new landmark or a landmark that was previously integrated in the map. This aspect was also studied in (Gil et al., 2008) and the u-SURF descriptor was the best choice (Bay et al., 2006).

2.2 3D visual fastSLAM

Initially, as the robots builds the maps independently, each one uses an independent particle filter. Then, after fusing the local maps, the robots continue the map building by means of a joint particle filter. In this section, we concentrate on the first case, i.e., FastSLAM for a single robot. In order to build the map, each robot is equiped with a stereo camera system, which enables them to obtain relative measurements to landmarks detected in the environment. In addition, these landmarks are characterized by means of a descriptor using visual information. In this context, each robot explores the environment while it observes visual landmarks. At time t the robot performs an observation $z_t = (v_t, d_t)$, where $v_t = (X_c, Y_c, Z_c)$ is three dimensional vector relative to the left camera reference frame and d_t is the visual descriptor associated to the landmark. The map L is represented by a collection of N landmarks $L = \{l_1, l_2, \ldots, l_N\}$. Each landmark is represented as: $l_k = \{\mu_k, \Sigma_k, d_k\}$, where $\mu_k = (X_k, Y_k, Z_k)$ is a vector describing the position of the landmark red to a global reference frame, with associated covariance matrix Σ_k. Furthermore, each landmark l_k is differentiated from others by means of a descriptor d_k. The robot pose at time t is denoted as x_t and movement of the robot at time t as u_t. On the other hand, $x^t = \{x_1, x_2, \ldots, x_t\}$ represents the robot path until time t, $z^t = \{z_1, z_2, \ldots, z_t\}$ is the set of observations made by the robot until time t and the set of actions $u^t = \{u_1, u_2, \ldots, u_t\}$. The solution to the SLAM problem consists in determining the location of the landmarks in the map L and the robot poses x^t from a set of measurements z^t and robot actions u_t. The set of data association performed until time t is denoted as $c^t = \{c_1, c_2, \ldots, c_t\}$. Each time the robot performs an observation, it has to determine whether this observation corresponds to a landmark previously incorporated in the map ($c_t \in [1 \ldots N]$) or to a new one ($c_t = N + 1$). The main idea of the FastSLAM algorithm is that the SLAM problem can be separated into two main subproblems: the estimate of the trayectory of the robot and the estimate of the map (Montemerlo & Thrun, 2003). This can be expressed as:

$$p(x^t, L|z^t, u^t, c^t) = p(x^t|z^t, u^t, c^t) \prod_{k=1}^{N} p(l_k|x^t, z^t, u^t, c^t) \quad (1)$$

This equation states that the SLAM posterior is decomposed into two parts: the estimate of the robot path and N independent estimators of the landmark positions, each conditioned to the path estimate. We approximate $p(x^t|z^t, u^t, c^t)$ by means of a set of M particles. Thus, each particle has N independent landmark estimators (implemented as EKFs), one for each landmark. Each particle is therefore defined as:

$$S_t^{[m]} = \{x^{t,[m]}, \mu_{t,1}^{[m]}, \Sigma_{t,1}^{[m]}, d_1^{[m]}, \ldots, mu_{t,N}^{[m]}, \Sigma_{t,N}^{[m]}, d_N^{[m]}\}, \quad (2)$$

where $\mu_{t,k}^{[m]}$ is the best estimation at time t for the position of landmark l_k based on the path of the particle m and $\Sigma_{t,k}^{[m]}$ the associated covariance matrix. The visual descriptor associated to the landmark j is represented by $d_j^{[m]}$. The particle set $S_t = \{S_t^{[1]}, S_t^{[2]}, \ldots, S_t^{[M]}\}$ is calculated incrementally from the set S_{t-1} in time $t-1$ and the control u_t. Each particle is sampled from a proposal distribution $x_t^{[m]} \sim p(x_t|x_{t-1}, u_t)$ that models the noise in the odometry of the robot. Then, a weight is assigned to each particle according to (Montemerlo & Thrun, 2003):

$$\omega_t^{[m]} = \frac{1}{\sqrt{|2\pi Z_{c_t}|}} e^{\{-\frac{1}{2}(v_t - \hat{v}_{t,c_t})^T [Z_{c_t}]^{-1} (v_t - \hat{v}_{t,c_t})\}} \tag{3}$$

where v_t is the actual measurement and $\hat{v}_{t,c_t}$ is the predicted measurement for the landmark c_t based on the pose $x_t^{[m]}$. The matrix Z_{c_t} is the covariance matrix associated with the innovation $(v_t - \hat{v}_{t,c_t})$. Note that it is assumed that each measurement v_t has been associated to the landmark c_t of the map. In short, the algorithm proposed in this paper is based on a Rao-Blackwellized particle filter that enables the robot to build a map and localize in an environment by means of the detection of three dimensional visual features. This algoritm can be decomposed in four basic steps:

a. Generating a new particle set.

b. Updating the estimate of the landmarks based on the observations.

c. Calculating a weight for each particle.

d. Resampling based on the weight of each particle.

In the following, we describe the previous steps of the FastSLAM algorithm.

2.2.1 New particle set generation

In the first step a new set of hypothesis S_t is obtained based on teh set S_{t-1}. That is to say, we obtain a new pose $x_t^{[m]}$ for the robot by sampling from a motion model:

$$x_t^{[m]} \sim p(x_t|x_{t-1}, u_t), \tag{4}$$

where $p(x_t|x_{t-1}, u_t)$ defines the movement model of the robot. This movement model is applied to each of the poses of each particle separately, based on the movement performed by the robot. Figure 1 shows the application of Equation 4. The figures show the path of three robots that work independently. Solid lines represent the real path followed by the robots, whereas dashed lines represent the odometry readings. It can be observed that, at the begining, all the particles are concentrated on the same point. After applying the movement model, we obtain a set of particle clouds that represent the probability of the pose of each robot at each time step.

2.2.2 Landmark estimate

The landmark update is performed based on the pose of the robot that performed the observation $z_t = \{v_t, d_t\}$ with data association c_t. This is done independently for each landmark l_k by means of the standard *EKF* equations presented below:

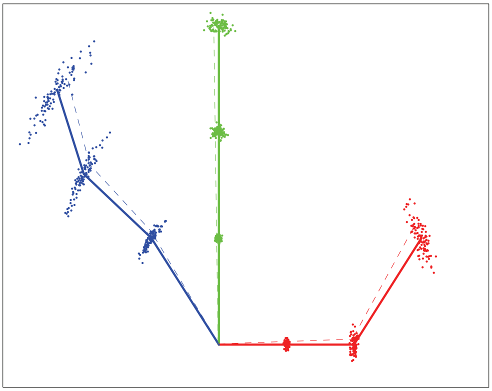


Fig. 1. The figure shows a new set of particles generated by sampling from the motion model.

$$\hat{v}_t = g(x_t^{[m]}, \mu_{c_t,t-1}^{[m]}) \tag{5}$$
$$G_{l_{c_t}} = \nabla_{l_{c_t}} g(x_t, l_{c_t})_{x_t = x_t^{[m]}; l_{c_t} = \mu_{c_t,t-1}^{[m]}} \tag{6}$$
$$Z_{c_t,t} = G_{l_{c_t}} \Sigma_{c_t,t-1}^{[m]} G_{l_{c_t}}^T + R_t \tag{7}$$
$$K_t = \Sigma_{c_t,t-1}^{[m]} G_{l_{c_t}}^T Z_{c_t,t}^{-1} \tag{8}$$
$$\mu_{c_t,t}^{[m]} = \mu_{c_t,t-1}^{[m]} + K_t(v_t - \hat{v}_t) \tag{9}$$
$$\Sigma_{c_t,t}^{[m]} = (I - K_t G_{l_{c_t}}) \Sigma_{c_t,t-1}^{[m]} \tag{10}$$

where $\hat{v}_t$ is the prediction for the current measurement v_t assuming that it has been associated with landmark c_t in the map. The observation model $g(x_t, l_{c_t})$ is linearly approximated by the Jacobian matrix $G_{l_{c_t}}$. It is assumed here that the noise in the observation is Gaussian and can be modeled with the covariance matrix R_t. Equation (9) represents the update of the estimate of the landmark c_t: $\mu_{c_t,t-1}^{[m]}$ based on the innovation $v = (v_t - \hat{v}_t)$. Finally, Equation (10) updates the covariance matrix $\Sigma_{c_t,t}^{[m]}$, which is associated to the m particle and the landmark c_t. Note that we implicitly assume that the observation z_t corresponds to the landmark l_{c_t} in the map. By now we assume this correspondence to be known.

2.2.3 Asigning weights to the particles and resampling

As seen in Section 4, the set of particles is generated by the movement model and distributed according to $p(x_t|x_{t-1}, u_t)$, which is known as proposal distribution. However, our aim is to estimate the posterior $p(x^t, L|z^t, u^t, c^t)$ which includes all the information from odometry and sensors until time t. This is known as target distribution. The difference between the proposal distribution and the target distribution is corrected with a process denoted as importance resampling (SIR). This process works as follows. A weight is assigned to each particle based on the quality of the current observation and the map matches. Then, a new set of particles S_t is created by sampling from S_{t-1}. Each particle is included in the new set with probability proportional to its weight. Assuming that a robot performs a single measurement z_t with data association c_t, the weight $w_t^{[m]}$ associated with the particle m as:

$$\omega_t^{[m]} = \frac{1}{\sqrt{|2\pi Z_{c_t}|}} e^{\{-\frac{1}{2}(v_t - \hat{v}_{t,c_t})^T [Z_{c_t}]^{-1} (v_t - \hat{v}_{t,c_t})\}} \tag{11}$$

Then, the weights are normalized to approximate a probability density function $\sum_{i=1}^{M} \omega_t^{[i]} = 1$. In this way, the set of particles represent a set of hipothetical paths that the robot may follow. Conditioned to each path, there is a set of 3D estimated landmarks, each one represented by a Kalman filter. Finally, in order to choose the path and map that best represents the true trajectories and environment, a logartihmic sum over the global weights of the particles is maintained. The most probable particle is selected as follows:

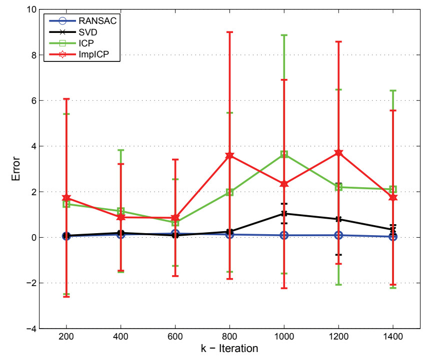
$$\hat{m} = \underset{m}{argmax} \sum_{t=1}^{A} \log(w_t^{[m]}) \tag{12}$$

3. Map fusion

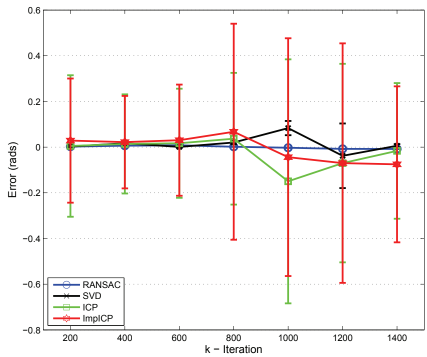
In the previous section we have seen how each robot is able to estimate a map of the environment by means of a particle filter. This task is initially performed individually. The result is a set of local maps whose landmarks are red to each robot system. At a specific moment, the fusion of these local maps into a global map may be required. In this section we concentrate on the map merging problem. This task is tackled in two subproblems: map alignment and map merging. These steps are described in detail in the following.

3.1 Map alignment

Aligning two maps means estimating the transformation between those maps. Concretely, three parameters are computed: a translation in x and y (t_x, t_y) and a rotation θ. In our case, each map is refered to the local reference system of the robot, which is located in its origin. In addition, the maps are landmark-based, thus the alingment is obtained by looking for correspondences based on the descriptor simmilarity. Afterwards, the landmarks of different maps can be expressed in the same reference system. In a previous work (Ballesta, Reinoso, Gil, Payá & Juliá, 2010), we studied, in detail, the problem of aligning three dimensional landmark-based maps. For this purpose, we compared a set of aligning methods suitable for this kind of maps. Particularly, we evaluated the following methods: RANSAC (RANdom SAmple Consensus), SVD (Singular Value Decomposition), ICP (Iterative Closest Point) and ImpICP (Improved ICP). The last one is an *ad-hoc* implementation. The experiments were carried out with simulated data as well as real maps built by the robots. Figures 2(a) and 2(b) show the results obtained. The experiments consisted of testing the behaviour of the aligning methods at different stages of the SLAM process. That is to say, initially the maps are smaller and obtaining the alignment is therefore more difficult. However, as the size of the maps is bigger, the common landmarks between those maps may increase, and thus the aligment can be succesfully found. This is represented in the figures that show the error obtained in the estimate of the aligning parameters. Figure 2(a) shows the error in the estimate of the translation parameters (t_x, t_y) when the alignment is obtained. Analogously, Figure 2(b) show the error in rotation (θ). Paying attetion to both figures, it is noticeable that RANSAC obtains the best results. The error is practically zero independently of the size of the maps. An example of the performance of RANSAC is illustrated in Figure 3. This figure shows how the common landmarks are indentified and matched.



(a)



(b)

Fig. 2. (a) Traslation error. (b) Rotation error.

3.2 Map merging

In this section we focus on the map merging problem. Once the alignment is performed, we will have the local maps expressed in the same ence system and the common landmarks between these maps identified. In this situation the local maps can be merged into a global one. As described previously, the maps built by the robots consist of a set of landmarks $L = \{l_1, l_2, \ldots, l_N\}$. Each landmark l_k is defined as $l_k = \{\mu_k, \Sigma_k, d_k\}$, where $\mu_k = (X_k, Y_k, Z_k)$ represents the position of the landmark by means of a three dimensional vector. Then Σ_k and d_k are the covariance matrix and the descriptor associated to each landmark. When merging two maps, the uncertainty in the estimate of the landmarks should be considered. We therefore propose a *Multivariable Stationary Kalman Filter* in order to fuse this data. Given two maps, map_1 and map_2, and using the nomenclature described previously, the following formulation is used to merge these maps:

$$K_{\{i\}} = \Sigma_{1\{i\}} \cdot (\Sigma_{1\{i\}} + \Sigma_{2\{i\}})^{-1} \tag{13}$$

$$\mu_{G\{i\}} = \mu_{1\{i\}} + K_{\{i\}} \cdot (\mu_{1\{i\}} - \mu_{2\{i\}}) \tag{14}$$

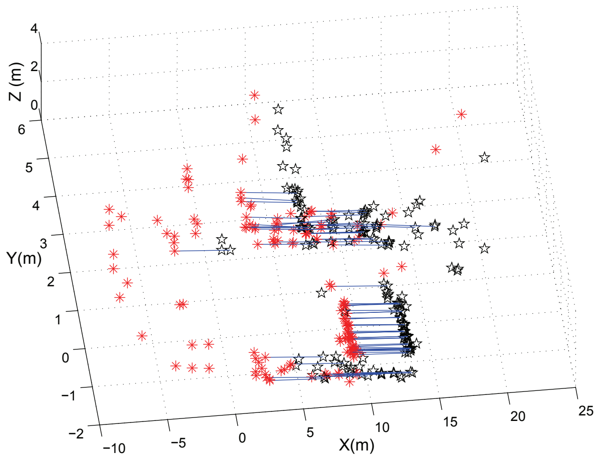


Fig. 3. Example of alignment.

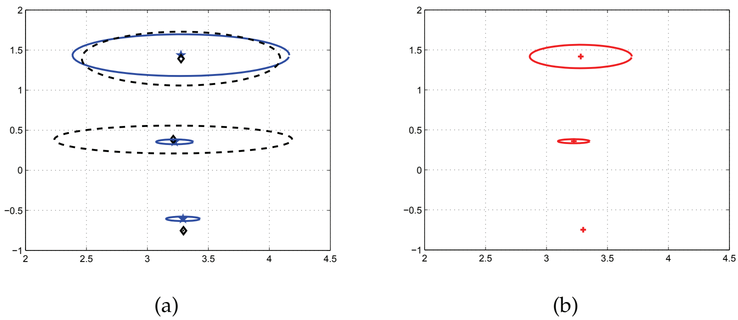


Fig. 4. (a) Position and uncertainty of 3 aligned landmarks from map_1 and map_2 (b) Same landmarks after fusion. Uncertainty is represented by an ellipse.

$$\Sigma_{G\{i\}} = (I - K_{\{i\}}) \cdot \mu_{1\{i\}} \tag{15}$$

where i is an index ($i \in \{1, N\}$) that denotes each matched landmark. N is the total number of matched landmarks between both maps (1 and 2). K_i represents the Kalman gain. $\mu_G\{i\}$ indicates the 3D coordinates of landmark i in the global map. This landmark is the result of matching and merging a common landmark between both local maps, map_1 and map_2. μ_1 are the 3D coordinates of map_1 and μ_2 the 3D coordinates of map_2 expressed in the map_1's reference system (i.e. after the alignment process). Finally, Σ_G,Σ_1 and Σ_2 are the 3×3 covariance matrices, which represent the uncertainty in the location of the landmarks in map_G, map_1 and map_2 respectively. The covariance matrices of map_2 (Σ_2) have been also transformed to the map_1's reference system. In Fig. 4, the uncertainty of the landmarks are represented with an ellipse. It can be observed that in the alignment process, we transform not only the position of the landmarks but also the error ellipse. This is done by means of a rotation matrix as follows:

$$\Sigma_2 = R^T \cdot \Sigma_{20} \cdot R \tag{16}$$

where Σ_{20} is the covariance matrix of map_2 before being aligned with map_1. And R is the transformation matrix which is:

$$R = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{17}$$

Finally, as far as the visual descriptor is concerned, we computed the mean between the descriptors in the local maps and incorporate the resulting descriptor to the global map.

4. Multirobot SLAM

Once the global map is obtained and the relative positions of the robots is known, the SLAM problem is solved jointly. That is to say, from the fusion of the local maps on, the robots perform the map building together. To do this, we propose the Rao-Blackwellized Kalman Filter extended to the multi-robot case. In this sense, the equation of the SLAM posterior is:

$$\begin{aligned} & p(x^t_{\langle 1:K\rangle}, L | z^t_{\langle 1:K\rangle}, u^t_{\langle 1:K\rangle}, c^t) = \\ & p(x^t_{\langle 1:K\rangle} | z^t_{\langle 1:K\rangle}, u^t_{\langle 1:K\rangle}, c^t) \prod_{k=1}^{N} \quad p(l_k | x^t_{\langle 1:K\rangle}, z^t_{\langle 1:K\rangle}, u^t_{\langle 1:K\rangle}, c^t) \end{aligned} \tag{18}$$

This probability function is a way to estimate a set of K paths $x^t_{\langle 1:K\rangle}$ and a map L conditioned to the case in which the robots perform a number of movements $u^t_{\langle 1:K\rangle}$ and a series of observations $z^t_{\langle 1_K\rangle}$ associated to landmarks in the map c^t. As it can be observed this expression is analogous to the equation 1, so the estimate of the map and the estimate of K paths can be separated into two parts: $p(x^t_{\langle 1:K\rangle} | z^t_{\langle 1:K\rangle}, u^t_{\langle 1:K\rangle}, c^t)$, which is estimated using a particle filter and the map L which is estimated using N independent estimates conditioned to the paths $x^t_{\langle 1:K\rangle}$. Analogously to Equation 2, the estimate of the posterior is done by means of M particles, each one represented as:

$$S_t^{[m]} = \{x^{t,[m]}_{\langle 1:K\rangle}, \mu^{[m]}_{1,t}, \Sigma^{[m]}_{1,t}, d^{[m]}_1, \cdots, \mu^{[m]}_{N,t}, \Sigma^{[m]}_{N,t}, d^{[m]}_N\} \tag{19}$$

Unlike the particle defined in (2), in this case the state that we would like to estimate is composed by the pose (x, y, θ) of K robots, thus $x^t_{\langle 1:K\rangle} = \{x_{t,\langle 1\rangle}, x_{t,\langle 2\rangle}, \cdots, x_{t,\langle K\rangle}\}$. As a result, we propose a joint estimation over a path state of dimension $3K$. According to (Thrun et al., 2005), the number of particles needed to obtain a good estimation increases exponentially with the dimension of the state. However, the results that we present here show that the approach works perfectly for robot teams of 2–3 members using a reasonable number of particles. In the case presented, the same map is shared by all the robots, which means that an observation performed by a particular robot affects the map of the whole robot team. For example, a robot does not need to explicitly close a loop to reduce the uncertainty in its pose. On the contrary, the robot can reduce its uncertainty if it observes landmarks previously seen by other robots. In consequence, one member of the team may observe a landmark previously mapped by a different robot and update its estimate. The formulation of the multi-robot FastSLAM algorithm proposed here is presented in algorithm 1.

Algorithm 1 Summary of the proposed algorithm.

$S = \emptyset$
$[z_{t,\langle 1\rangle}, z_{t,\langle 2\rangle}, z_{t,\langle 3\rangle}]$ = ObtainObservations()
InitialiseMap$(S, x_{0,\langle 1:3\rangle}, z_{t,\langle 1:3\rangle})$
for $t = 1$ to numMovements **do**
  $[z_{t,\langle 1\rangle}, z_{t,\langle 2\rangle}, z_{t,\langle 3\rangle}]$ = ObtainObservations()
  $[S, \omega_{t,\langle 1\rangle}]$ = FastSLAMMR$(S, z_{t,\langle 1\rangle}, R_t, u_{t,\langle 1\rangle})$
  $[S, \omega_{t,\langle 2\rangle}]$ = FastSLAMMR$(S, z_{t,\langle 2\rangle}, R_t, u_{t,\langle 2\rangle})$
  $[S, \omega_{t,\langle 3\rangle}]$ = FastSLAMMR$(S, z_{t,\langle 3\rangle}, R_t, u_{t,\langle 3\rangle})$
  $\omega_t = \omega_{t,\langle 1\rangle}\omega_{t,\langle 2\rangle}\omega_{t,\langle 3\rangle}$
  S = ImportanceResampling(S, ω_t) //Sample randomly from S according to $\omega_t^{[m]}$
end for

function $[S_t]$ **= FastSLAMMR**$(S_{t-1}, z_{t,\langle i\rangle}, R_t, u_{t,\langle i\rangle})$
$S_t = \emptyset$
for $m = 1$ to M {For every particle} **do**
  $x_{t,\langle i\rangle}^{[m]} \sim p(x_{t,\langle i\rangle} | x_{t-1,\langle i\rangle}, u_{t,\langle i\rangle})$
  for $n = 1$ to $N_{t-1}^{[m]}$ //Loop over all possible data associations **do**
    $\hat{v}_{t,\langle i\rangle} = g(x_{t,\langle i\rangle}^{[m]}, \mu_{n,t-1}^{[m]})$
    $G_{l_n} = \nabla_{l_{c_t}} g(x_t, l_n)_{x_{t,\langle i\rangle} = x_{t,\langle i\rangle}^{[m]}; l_{c_t} = \mu_{c_t,t-1}^{[m]}}$
    $Z_{n,t} = G_{l_n} \Sigma_{n,t-1}^{[m]} G_{l_n}^T + R_t$
    $D(n) = (v_{t,\langle i\rangle} - \hat{v}_{t,\langle i\rangle})^T [Z_{n,t}]^{-1} (v_{t,\langle i\rangle} - \hat{z}_{t,\langle i\rangle})$
    $E(n) = (d_{t,\langle i\rangle} - d_n)^T (d_{t,\langle i\rangle} - d_n)$
  end for
  $D(N_{t-1}^{[m]} + 1) = D_0$
  $j = find(D \leq D_0)$ {Find candidates below D_0}
  $c_t = argmin_j E(n)$ {Find minimum among candidates}
  if $E(c_t) > E_0$ // Create a new landmark? **then**
    $c_t = N_{t-1}^{[m]} + 1$
  end if
  if $c_t = N_{t-1}^{[m]} + 1$ //New landmark **then**
    $N_t^{[m]} = N_{t-1}^{[m]} + 1$
    $\mu_{c_t,t}^{[m]} = g^{-1}(x_{t,\langle i\rangle}^{[m]}, z_{t,\langle i\rangle})$
    $\Sigma_{c_t,t}^{[m]} = G_{l_{c_t}}^T R_t^{-1} G_{l_{c_t}}$
    $\omega_t^{[m]} = p_0$
  else
    $N_t^{[m]} = N_{t-1}^{[m]}$ //Old landmark
    $K_t = \Sigma_{c_t,t-1}^{[m]} G_{L_{c_t}}^T Z_{c_t,t}^{-1}$
    $\mu_{c_t,t}^{[m]} = \mu_{c_t,t-1}^{[m]} + K_t(v_{t,\langle i\rangle} - \hat{v}_{t,\langle i\rangle})$
    $\Sigma_{c_t,t}^{[m]} = (I - K_t G_{l_{c_t}}) \Sigma_{c_t,t-1}^{[m]}$
  end if
  $\omega_{t,\langle i\rangle}^{[m]} = \frac{1}{\sqrt{|2\pi Z_{c_t}|}} e^{\{-\frac{1}{2}(v_{t,\langle i\rangle} - \hat{v}_{t,c_t})^T [Z_{c_t}]^{-1} (v_{t,\langle i\rangle} - \hat{v}_{t,c_t})\}}$
  add $\{x_{t,\langle i\rangle}^{[m]}, N_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_{1,t}^{[m]}, \cdots, \mu_{N_t^{[m]},t}^{[m]}, \Sigma_{N_t^{[m]},t}^{[m]}, d_{N_t^{[m]},t}^{[m]}, \omega_t^{[m]}\}$ to S_t
end for
return S_t

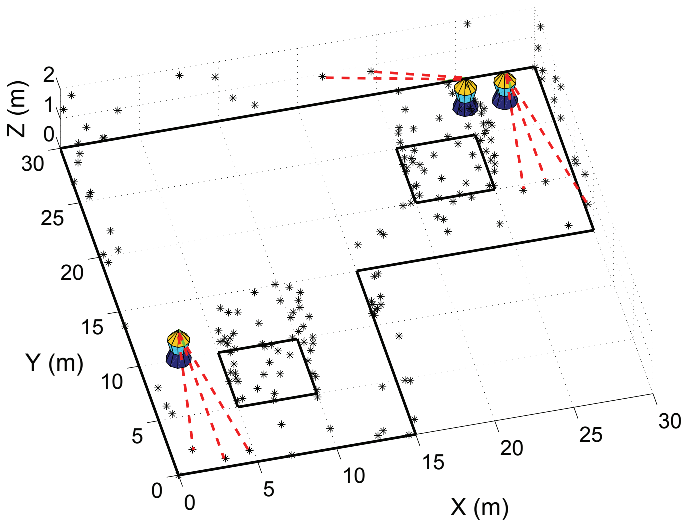


Fig. 5. Simulated environment. The robots perform observations of simulated landmarks located in walls. These walls are represented by a line. It can be observed that the walls restrict the visibility of the visual landmarks placed behind.

5. Experiments

In order to test the SLAM algorithm proposed, we have created a simulated environment as shown in figure 5. In this figure there are three robots building a map of the environment. This environment is represented by walls (line) with landmarks randomly located in them. In the figure we can see which landmarks are the robots currently observing. This is represented by dashed lines. In the experiments performed we have test the computational cost of the algorithm as well as the RMS error in the estimate of the error. Regarding the computational time, two parameteres are mainly influencing: the number of particles used (M) and the number of observations integrated as each time step (B). In order to decrease the computational cost, we can reduce the number of observations that each robot reduces at each iteration of the algorithm. For example, we consider that each robot obtains B/K observations. That is to say, if we had three robots and $B = 15$, then each robot would integrate $B = 5$ observations. Figure 6(a) shows the results obtained if robots perform B/K observations. In this case, it can be observed that the computational cost is simmilar regardless of the number of robots, for any number of particles. The total number of observations integrated in the filter is the same for the case of 1, 2 or 3 robots. On the other hand, in Figure 6(b) we evaluate the error in the map, using the previous restriction (B/K observations). As the figure shows, the estimate of the map and the paths improves when more robots are used. It can be deduced, from these figures, that good results can be obtained even if we reduce the number of observations performed by each robot.

6. Conclusion

This paper presents a posible solution to the multi-robot SLAM problem in the visual context. Particularly, the approach proposed begins with an independent map building part, in which

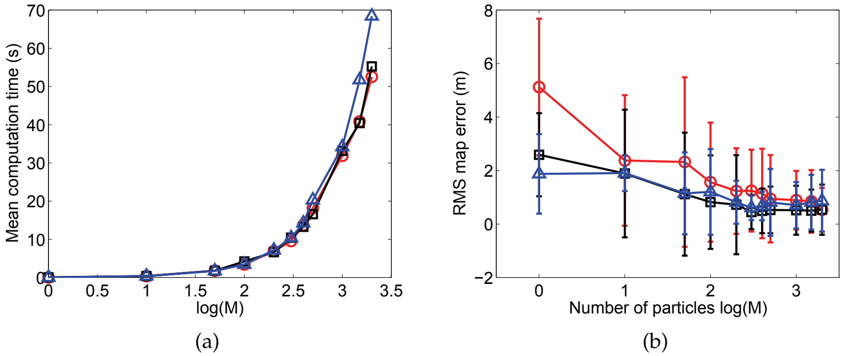


Fig. 6. Figure (a) shows the mean computation time at each iteration of the algorithm for different number of particles $M = \{1\ 10\ 50\ 100\ 200\ 300\ 400\ 500\ 1000\ 1500\ 2000\}$. Figure (b) shows the RMS error in the map when B/K observations are used. We show simultaneously results with one robot (◯), two robots (□) and three robots (△).

each robot builds its own map independently. In this case, the relative positions of the robots are not necessary. This is an advantage in practice, since this data could be unknown. As a consequence, the robots build their own local maps regardless of other robots' poses and observations. The map building is carried out by means of a Rao-Blackwellized particle filter. Since the robots work independently, this algorithm has been implemented for a single robot. At some point, the robots may share the information collected and fused the local maps into a single one. For this reason, the map fusion problem has been tackled in this paper. First, we paid attention to the alignment problem in which a common reference system for the local maps is obtained. Then, we concentrated on the map merging problem. In this case, the common landmarks are identified based on the descriptor similarity and fused, taking into account the uncertainty in the estimate of those landmarks. To do this, we use a *Multivariable Stationary Kalman Filter*. The results show that the uncertainty of the fused landmarks is reduced. Finally, once the relative positions of the robots are known and a global map is computed, the SLAM process is continued by means of an extension of the previous particle filter. This time, we implemented a multi-robot particle filter in such a way that the robots estimate their trajectories and the map jointly. The results obtained show the good performance of the algorithm in simulation. As future work, it is desirable to evaluate this algorithm in a real environment.

7. References

Ballesta, M., Gil, A., Reinoso, O. & Úbeda, D. (2010). Análisis de detectores y descriptores de características visuales en slam en entornos interiores y exteriores, *Revista Iberoamericana de Automática e Informática Industrial (RIAII)* 7(2): 68–80.

Ballesta, M., Reinoso, O., Gil, A., Payá, L. & Juliá, M. (2010). Evaluation of aligning methods for landmark-based maps in visual slam, *INTECH* . To appear.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf: Speeded up robust features, *Proceedings of the ninth European Conference on Computer Vision*.

Davison, A. J. & Murray, D. W. (2002). Simultaneous localisation and map-building using active vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Fenwick, J. W., Newman, P. M. & Leonard, J. J. (2002). Cooperative concurrent mapping and localization, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* pp. 1810–1817.

Gil, A., Martínez Mozos, O., Ballesta, M. & Reinoso, O. (2009). A comparative evaluation of interest point detectors and local descriptors for visual slam, *Machine Vision and Applications Journal* .

Gil, A., Martínez-Mozos, ., Ballesta, M. & Reinoso, . (2008). A comparative evaluation of interest point detectors and local descriptors for visual slam, *Machine Vision and Applications* .

Gil, A., Reinoso, O., Fernández, C., Vicente, M. A., Rottmann, A. & Martínez-Mozos, O. (2006). Simultaneous localization and mapping in unmodified environments using stereo vision, *Proceedings of the 3rd International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal.

Gil, A., Reinoso, O., Martínez-Mozos, O., Stachniss, C. & Burgard, W. (2006). Improving data association in vision-based SLAM, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China.

Harris, C. G. & Stephens, M. (1998). A combined corner and edge detector, *Alvey Vision Conference*.

Jensfelt, P., Kragic, D., Folkesson, J. & Björkman, M. (2006). A framework for vision based bearing only 3D SLAM, *IEEE Int. Conf. on Robotics & Automation*, Orlando, FL, USA.

Konolige, K., Fox, D., Limketkai, B., Ko, J. & Stewart, B. (2003). Map merging for distributed robot navigation, *Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Kwak, N., Kim, G.-W., Ji, S.-H. & Lee, B.-H. (2008). A mobile robot exploration strategy with low cost sonar and tungsten-halonen structural ligth, *Journal of Intelligent and Robotic Systems* 51(1): 89–111.

Leonard, J. J. & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons, *IEEE Transactions on Robotics and Automation* 7(4).

Little, J., Se, S. & Lowe, D. (2002). Global localization using distinctive visual features, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Lausanne, Suiza.

Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Montemerlo, M. & Thrun, S. (2003). Simultaneous localization and mapping with unknown data association using FastSLAM, *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Taipei, Taiwan.

Murillo, A. C., Guerrero, J. J. & Sagüés, C. (2007). Surf features for efficient robot localization with omnidirectional images, *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, San Diego, CA, USA.

Murray, D. & Little, J. J. (2000). Using real-time stereo vision for mobile robot navigation, *Autonomous Robots* 2(8): 161–171.

Roumeliotis, S. & Bekey, G. (2002). Distributed multi-robot localization, *IEEE Transactions on Robotics and Automation* 18(5): 781–795.

Stewart, B., Ko, J., Fox, D. & Konolige, K. (2003). A hierarchical bayesian approach to mobile robot map structure estimation, *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico.

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots, *Int. Journal of Robotics Research* 20(5): 335–363.

Thrun, S., Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, The MIT Press. ISBN: 0-262-20162-3.

Valls-Miró, J., Zhou, W. & Dissanayake, G. (2006). Towards vision based navigation in large indoor environments, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China.

Wijk, O. & Christensen, H. I. (2000). Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data, *Robotics and Autonomous Systems* 1(31): 31–42.

Zhou, X. S. & Roumeliotis, S. I. (2006). Multi-robot slam with unknown initial correspondence: The robot rendezvous case, *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 1785-1792*.