

Cooperative SLAM using Independent Rao-Blackwellized Filters

Abstract

This paper focusses on an approach to the multi-robot SLAM problem. We consider that a team of robots start the map building process from known initial poses and move through the environment observing a set of landmarks. In particular, we consider that the robots are equipped with vision sensors and are capable to compute the relative distance to natural visual landmarks. In the presented approach, each robot computes an own global map and integrates its own observations in it. In addition, each robot is also capable of integrating the measurements obtained by other robots in his own map, thus cooperating to compute a global map using the information provided by all the robots in the team. In this way, each robot in the team builds a different global map. The approach presented here differs from others in the sense that each robot maintains an independent SLAM filter, and still each robot can introduce the observations of other robots in his own map. We present a set of experimental results obtained in a simulated environment that validate the proposed approach.

Index Terms

multi-robot SLAM, visual SLAM, particle Filter

I. INTRODUCTION

Map construction is a typical mission that autonomous mobile robots should be capable to accomplish, since often an accurate model of the environment is needed for navigation [5]. In order to build the map, we consider the situation in which a group of autonomous mobile robots move through an unknown space and incrementally build a map of it, while, simultaneously, use this map to compute their absolute location. This case can be considered as an extension of the Simultaneous Localization and Mapping (SLAM) problem which has attracted the attention of scholars during the last two decades [15], [30]. The SLAM problem is considered hard, since any error in the estimation of the robot pose will lead to an error in the estimation of the map and *viceversa*.

Some applications require the use of several robots that work together in the environment. The necessity of using multiple robots can be justified, since multiple vehicles can frequently perform tasks more quickly

and robustly than a single one [17], [23]. However, the SLAM problem becomes harder in this case, since the trajectories of several robots need to be estimated, while, simultaneously fusing the information from different platforms to estimate a single coherent map. As a result, the dimensionality of the problem is increased.

Regarding the sensors that the robots use to extract information from the environment, to date, many approaches used range sensors such as SONAR [14], [29] or laser [25], [27]. Others combine different sensor sources for map building [21]. Lately, there exists an increasing interest on using cameras as sensors for SLAM. This approach is denoted as visual SLAM [11], [28], since the robots use visual information in order to build a model of the environment. Cameras offer a higher amount of information from the environment and are commonly less expensive than laser sensors. We concentrate on the case in which the mobile robots are equipped with a stereo vision sensor, that allows to compute the relative 3D coordinates of the points from the viewed scene [7]. [Other authors propose the use of omni-directional sensor, which maximize the field of view of the robot, however the depth estimation in this case is usually difficult \[10\].](#) Most approaches to visual SLAM are feature-based. In this case, a set of salient points in the environment are used as landmarks [24]. Mainly, two steps must be distinguished in the selection of visual landmarks. The first step involves the detection of interest points in the images that can be used as reliable landmarks. The points should be detected at different distances and viewing angles, since they will be observed by the robot from different poses in the environment. At a second step the interest points are described by a feature vector, which is computed using local image information. To sum up, we consider that each robot is equipped with a stereo camera and is able to obtain 3D relative measurements to visual landmarks accompanied by a descriptor. [It is worth noting that we do not assume the existence of any artificial landmark in the environment \(such as, for example, circle-like structures \[6\]\) and consider any kind of natural landmark. Thus, no modification is needed to be performed in the environment.](#)

The data association problem is in close relation with the SLAM problem. Given an observation obtained by the robot, it has to be associated to one of the landmarks in the map. In this sense, the robot has to decide whether the observation corresponds to one of the landmarks previously integrated in the map, or, on the contrary, it is a new one. We propose to use the visual description associated to the landmark along with the distance measurement obtained by the robot in the data association problem. Typically, when the robot traverses already explored places, it detects previously observed landmarks. In this case, if the current observations are correctly associated with the visual landmarks in the map, the robot will be able to find its location with respect to these landmarks, thus reducing the uncertainty in its pose. Moreover, when different robots cooperate to build a single map of the environment, the data

association must be solved taking into account the observations of the different agents that participate in the task. In this way, a robot should be capable of associating its current observations with the landmarks previously seen by other robots, thus localizing in a map area previously explored by a different robot.

It is worth noting that SLAM algorithms focus on the incremental construction of a map, given a set of movements carried out by the robots and the set of observations obtained from different locations. In consequence, we do not consider the computation of the movements that need to be performed by the robots, since this is generally considered a different problem, denoted as exploration.

The main contribution of this paper consists in an approach to the multi-robot visual SLAM problem using a Rao-Blackwellized Particle Filter (RBPF). To date, some solutions have employed a RBPF to estimate a map using several robots [9], [12], [20]. In these approaches a unique particle filter is employed to compute a single map in a centralized manner, where each particle in the filter describes the path of all the robots in the team, and a map conditioned to them. With this approach, the dimensions of the state to estimate grow rapidly with the number of robots in the team, thus requiring a large number of particles during the estimation. In consequence, these solutions suffer from scalability problems. Other solutions propose the estimation of an own individual map using each robot's observations independently [13], [31]. Later, the local maps are aligned and fused to compute a common map [1], [2]. When using this approach, new observations should only be compared to a limited number of landmarks in the local map. Additionally, the construction of the local maps can be performed even if the relative positions of the robots are unknown. However, the main problem in this kind of solutions is that there exists no cooperation in the map building task, since the robots do not share the observations. This means that a robot may not localize in a place already visited by a different robot.

Opposite to these solutions, in the approach presented here each robot maintains an independent particle filter. As we will show, maintaining independent filters is advantageous, for several reasons, such as robustness and computational effort when compared to solutions using a single unified filter for the SLAM process. In our approach, each robot builds an own global map using the information of its own sensor and, in addition, integrates the information gathered by the other robots in the map. In order to do this, the relative initial position of each robot in the beginning is known. Since each robot in the team builds a global map, the system will be robust against the failure of a single member. It is worth noting that each robot is able to close the loop when it travels through an area previously explored by another robot.

The algorithm has been validated using simulated data in an office-like environment. The main reason that motivated to carry out a series of experiments using simulated data is that this enables to assess

the validity of the presented algorithm under different conditions. During exploration, the robots capture images from the simulated environment and extract SURF features from them [4]. Each visual landmark is identified by a SURF descriptor, used previously in robotic applications [19]. Other descriptors have been used to date [16], [22], however the SURF descriptors have shown outstanding results in the field of visual SLAM [8].

The remainder of the paper is structured as follows: Section II presents our approach to multi-robot SLAM. Following, Section III exposes our approach to the data association problem in the context of visual landmarks. Next, in Section IV we present the experimental results obtained. Finally, Section V summarizes the most important conclusions.

II. INDEPENDENT FASTSLAM

In this section we describe the approach to multi-robot SLAM. We will refer to this approach as iFastSLAM since it maintains an independent FastSLAM filter for each robot. We consider the case in which K mobile robots explore the environment while simultaneously build a map. Existing SLAM algorithms share this idea but differ in the methods used to solve this problem. In this paper the SLAM algorithm is based on a Rao-Blackwellized particle filter, denoted generally as FastSLAM [18]. This algorithm uses a particle set to represent the uncertainty of the robot's pose whereas each particle has its own associated map. The main idea of the FastSLAM algorithm is to separate the two fundamental aspects of the SLAM problem: the estimate of the robot's pose and the estimate of the map. The solution to the SLAM problem is performed by means of a sampling and particle generation process, in which the particles whose current observations do not fit with their associated map are eliminated.

To date, some authors have proposed an extension of the FastSLAM algorithm to the multi-robot case. In this case, they propose the estimation of the joint probability over the paths of all the robots in the team and the map. According to [9], [12], [20] the multi-robot SLAM problem can be stated as the estimation of the following probability function:

$$p(x_{1:t}^{(1:K)}, L | z_{1:t}^{(1:K)}, u_{1:t}^{(1:K)}, c_{1:t}) = p(x_{1:t}^{(1:K)} | z_{1:t}^{(1:K)}, u_{1:t}^{(1:K)}, c_{1:t}) \cdot \prod_{n=1}^N p(l_n | x_{1:t}^{(1:K)}, z_{1:t}^{(1:K)}, u_{1:t}^{(1:K)}, c_{1:t}), \quad (1)$$

where we denote the path of the robot $\langle i \rangle$ from time 1 to time t as $x_{1:t}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_t^{(i)}\}$. We aim at estimating the path of the whole robot team, which can be represented as: $x_{1:t}^{(1:K)} = \{x_{1:t}^{(1)}, x_{1:t}^{(2)}, \dots, x_{1:t}^{(K)}\}$. Analogously, we denote $u_{1:t}^{(1:K)} = \{u_{1:t}^{(1)}, u_{1:t}^{(2)}, \dots, u_{1:t}^{(K)}\}$ the set of actions carried out by the robots and $z_{1:t}^{(1:K)} = \{z_{1:t}^{(1)}, z_{1:t}^{(2)}, \dots, z_{1:t}^{(K)}\}$ refers to the set of measurements performed by the robot team until time t . We consider that, at a time step t the robot $\langle i \rangle$ is at pose $x_t^{(i)}$ and obtains an observation, constituted

by a measurement $z_t^{(i)}$ and a visual descriptor $d_t^{(i)}$. Finally, the map L is represented by N landmarks $L = \{l_1, l_2, \dots, l_N\}$, each one described as $l_k = \{\mu_k, \Sigma_k, d_k\}$, where $\mu_k = (X_k, Y_k, Z_k)$ is the mean position of the landmark, with associated covariance matrix Σ_k . In addition, each landmark l_k is associated with a SURF descriptor d_k .

This equation proposes a manner to estimate a group of K paths $x_{1:t}^{(1:K)} = \{x_{1:t}^{(1)}, x_{1:t}^{(2)}, \dots, x_{1:t}^{(K)}\}$ and a map L , conditioned to the case that the robots have performed a number of movements $u_{1:t}^{(1:K)}$ and a series of observations $z_{1:t}^{(1:K)}$ associated to landmarks in the map $c_{1:t}$. In consequence, Equation (1) expresses that we can separate the estimation of the map and the estimation of K different paths into two parts: The function $p(x_{1:t}^{(1:K)} | z_{1:t}^{(1:K)}, u_{1:t}^{(1:K)}, c_{1:t})$ is estimated using a particle filter, while the map is estimated using N independent estimations conditioned to the paths $x_{1:t}^{(1:K)}$. The main drawback of this idea is the number of particles needed to represent accurately the probability over the paths of all robots. According to [26] the number of particles required grows exponentially with the number of robots. In consequence, this solution suffers from scalability problems and may only be used for robot teams with 2–3 vehicles [9], [12], [20].

In contrast with the previously mentioned approaches, the algorithm presented in this paper maintains an independent filter for each of the robots, and however, is able to compute a global map using the observations of all the robots. The key idea of the approach presented here, is that any robot in the team builds an independent map and uses the rest of the robots as remote sensors. Each of the robots obtains its own measurements and integrates them in its own map, and in addition, each robot also receives the observations of other members. When each robot moves along the environment, it extracts landmarks with its onboard sensor and integrates them in its own global map and sends them to the rest of the robots. At each step, each robot also integrates in its own map the measurements of the other robots. In order to do this, the uncertainty of each robot is taken into account.

Since the filters are independent, the complexity of the algorithm grows linearly with the number of robots. Although the filters are independent, each of the robots is able to build a common map. In order to do this, we propose the estimation of a slightly different probability distribution function compared to Equation (1). We consider that the probability over the path of the robot $\langle i \rangle$ and map can be estimated as:

$$p(x_{1:t}^{(i)}, L | z_{1:t}^{(i)}, \bar{z}_{1:t}^{(o)}, u_{1:t}^{(1:K)}, c_{1:t}) = p(x_{1:t}^{(i)} | z_{1:t}^{(i)}, \bar{z}_{1:t}^{(o)}, u_{1:t}^{(1:K)}, c_{1:t}) \cdot \prod_{n=1}^N p(l_n | x_{1:t}^{(i)}, \bar{x}_{1:t}^{(o)}, z_{1:t}^{(i)}, \bar{z}_{1:t}^{(o)}, u_{1:t}^{(1:K)}, c_{1:t}), \quad (2)$$

where the notation $\langle o \rangle$ means other robots different from the current robot $\langle i \rangle$, that is $\langle o \rangle = \langle 1, \dots, i - 1, i + 1, \dots, K \rangle$. We denote as $\bar{x}_{1:t}^{(o)}$ the mean of the paths associated to other robots that are not the current

robot $\langle i \rangle$, which can be computed from the particle clouds associated to the other robots. In addition, we denote as $\bar{z}_{1:t}^{(o)}$ the observations seen by other robots transformed to the reference frame of robot $\langle i \rangle$ using the mean value of the particles associated to other robots.

We propose the estimation of Equation (2) instead of Equation (1) to solve the SLAM problem. Equation (2) represents the product of two terms. The first part $p(x_{1:t}^{(i)} | z_{1:t}^{(i)}, \bar{z}_{1:t}^{(o)}, u_{1:t}^{(1:K)}, c_{1:t})$ is the probability over the path of the robot $\langle i \rangle$ and will be estimated using a set of particles, whereas the second term expresses the product of a set of independent landmark estimators. The estimation of each landmark l_n is conditioned to the path of the robot $\langle i \rangle$ and to the mean path of the rest of the robots $\bar{x}_{1:t}^{(o)}$. That means that, in order to update the path and the associated map of the robot $\langle i \rangle$ at time t , we approximate the position and orientation of the other robots by a gaussian distribution. That is, we compute a normal distribution $N(\bar{x}_t^{(j)}, F_u^{(j)})$ that represents the particles associated to other robots. For example, a different robot $\langle j \rangle$ will be approximated with a gaussian distribution with mean $\bar{x}_t^{(j)}$ and covariance $F_u^{(j)}$ computed from its particle cloud. The computation of $\bar{x}_t^{(j)}$ and $F_u^{(j)}$ is detailed in the [Appendix](#). This approximation is, in general, not exact, since particle filters are able to represent multimodal densities. However, the results obtained show that in most of the cases, this approximation is feasible, leading to a reduced number of particles to estimate a joint map using the observations of all the robots in the team. Assuming this approximation, the second part in Equation (2) represents $p(L | x_{1:t}^{(i)}, \bar{x}_{1:t}^{(o)}, z_{1:t}^{(i)}, \bar{z}_{1:t}^{(o)}, u_{1:t}^{(1:K)}, c_{1:t})$ that indicates that the map L is still conditioned to the paths of the rest of the robots, thus it is statistically consistent.

On the contrary, other solutions based on the fusion of local maps do not take into account the uncertainty in the location of the vehicles when performing the fusion step [3], leading frequently to incoherent maps.

We maintain a separated particle set $S_t^{(i)}$ associated to every robot in the team with $i \in [1, K]$. Each particle set $S_t^{(i)}$ represents the path followed by the robot $\langle i \rangle$. Conditioned to this path, a map is computed based on the observations of the $\langle i \rangle$ robot and the rest of the robots, with the proposed gaussian approximation. Thus, we define the particle set associated to robot $\langle i \rangle$ as:

$$S_t^{[m]\langle i \rangle} = \{x_{1:t}^{[m]\langle i \rangle}, \mu_{t,1}^{[m]\langle i \rangle}, \Sigma_{t,1}^{[m]\langle i \rangle}, d_1^{[m]\langle i \rangle}, \dots, \mu_{t,N}^{[m]\langle i \rangle}, \Sigma_{t,N}^{[m]\langle i \rangle}, d_N^{[m]\langle i \rangle}\}. \quad (3)$$

where m represents the index of the particle, formed by a total of M particles.

In order to differentiate between the own measurements and the measurements performed by other robots, we will use the following notation: $z_t^{(i)}$ denotes the observations obtained with the onboard sensor or robot $\langle i \rangle$, whereas $\bar{z}_t^{(i,j)}$ denotes the measurement observed by the robot $\langle j \rangle$ that will be integrated in

the SLAM filter of robot $\langle i \rangle$. For example, $\bar{z}_t^{(1,2)}$ denotes an observation received by robot $\langle 1 \rangle$ that was observed initially by robot $\langle 2 \rangle$. The error associated to the measurement $\bar{z}_t^{(i,j)}$ is considered gaussian and is modelled by a $N(0, R_e^{(i,j)})$, where we define $R_e^{(i,j)}$ as the extended error matrix. This matrix propagates the error in the remote robot to the error in the $\langle i \rangle$ robot's reference frame. The covariance matrix $R_e^{(i,j)}$ considers the inner error of the remote sensor and also the uncertainty in the location of the remote robot $\langle j \rangle$ with respect to the robot $\langle i \rangle$.

Figure 1 illustrates the basic idea of the method presented here. In the figure, the position of two robots, Robot 1 and Robot 2 is drawn with triangles. In addition, the position of two landmarks is marked with stars. We consider that Robot 1 is only able to detect the leftmost landmark and obtains a measurement $z_t^{(1)}$, whereas Robot 2 detects the remaining landmark and computes an observation $z_t^{(2)}$. In this figure, the uncertainty of Robot 1 is represented by a set of particles, whereas, in the case of Robot 2, the set of particles is approximated by a gaussian distribution (dashed ellipse). The figure assumes that Robot 1 integrates an own observation $z_t^{(1)}$ and, in addition, it also integrates an observation only seen by Robot 2 $\bar{z}_t^{(o)}$ (dashed arrow). The computation of $\bar{z}_t^{(o)}$ considers the transformation to the reference system of Robot 1 and takes into account the uncertainty of Robot 2, approximated as a gaussian distribution. The computation $\bar{z}_t^{(o)}$ and the uncertainty $R_e^{(i,j)}$ associated to it is detailed in the Appendix.

The proposed method can be separated in three basic steps:

- Generate a new particle set based on the prior set.
- Update the estimation of each landmark based on the observations.
- Calculate a weight for each particle and perform a resampling based on the weight of each particle.

Each of these steps is detailed in the following subsections.

A. Generate a new particle set

Being K the number of robots, we maintain a different particle set $S_t^{(i)}$ associated to every robot in the team with $i \in [1, K]$. The first step is to generate a new set of hypotheses $S_t^{(i)}$ based on the set $S_{t-1}^{(i)}$. That means that we obtain a new particle set over the robot pose $x_t^{(i)}$. That is, we obtain a new pose $x_t^{[m](i)}$ for each of the robots by sampling from a motion model as indicated in [9], [18].

Since we maintain a different particle set for every robot in the team, each particle associated to the robot $\langle i \rangle$ is independent of any particle associated to robot $\langle j \rangle$.

B. Landmark estimation

In this step, we consider that the robot $\langle i \rangle$ received an observation and has associated this observation with the landmark l_{c_t} in his map. The procedure used to compute this correspondence is detailed in Section III. We have to distinguish two cases: first, if the current measurement was obtained by the robot $\langle i \rangle$, we have an observation $z_t^{(i)}$. Second, if the current measurement was performed by a different robot $\langle j \rangle$, we would like to integrate the observation $\bar{z}_t^{(i,j)}$ in the filter of robot $\langle i \rangle$. In the first case, we compute the noise matrix R_t using the equations that model the robot's onboard sensor. In the second case in order to integrate $\bar{z}_t^{(i,j)}$, we calculate an extended noise matrix $R_e^{(i,j)}$ that accounts for the uncertainty in the position of the robot $\langle j \rangle$ and the intrinsic error of the onboard sensor of robot $\langle j \rangle$. The computation of $\bar{z}_t^{(i,j)}$ and the extended noise matrix $R_e^{(i,j)}$ is based on a first order propagation law and is detailed in the Appendix. The update equation of each landmark estimate is based on the pose of the $\langle i \rangle$ robot. Each landmark in the map is updated independently, using the standard EKF equations as explained in [18].

C. Assigning a weight to each particle and Importance resampling

A weight $\omega_t^{[m](i)}$ associated with the particle m and the robot $\langle i \rangle$ is computed as [9], [18]. Basically, a weight is assigned to each particle depending on the quality that the current observation and the map match.

Next, the weights computed in the previous section are normalized to approximate a probability density function, so that $\sum_{m=1}^M \omega_t^{[m](i)} = 1$. Following, a new set of particles $S_t^{(i)}$ is created by sampling from $S_{t-1}^{(i)}$. Each particle is included in the new set with probability proportional to its weight.

III. DATA ASSOCIATION

While the robot explores the environment it must decide whether the measurement $z_t^{(i)}$ and visual descriptor $d_t^{(i)}$ correspond to a particular landmark in the map or to a new one. Note that the observation consists of a relative distance measurement $z_t^{(i)}$ and, since we are using a vision sensor, a descriptor $d_t^{(i)}$ that describes the visual appearance of the point in space. In this case, the robot must decide whether the measurement corresponds to one of the landmarks in the map $l_n, n \in [1, N]$, or it is a new landmark l_{N+1} . In order to decide the data association we follow the procedure indicated in [9].

The data association is performed independently for each particle, and this means that different particles would make different data associations. In addition, the same solution can be used for the observations coming from other robots different from the current robot $\langle i \rangle$. That is, if we receive an observation $\bar{z}_t^{(i,j)}$ with descriptor $d_t^{(i,j)}$ it will be associated to one of the landmarks in the map of robot $\langle i \rangle$. Thus, a robot

can localize with respect to landmarks seen by other robots in the team, thus the team cooperates to build the map.

IV. EXPERIMENTAL RESULTS

In this section we will present results that demonstrate the validness of the algorithm presented in Section II. We have tested the algorithm when using a different number of particles in the estimation and varying the trajectories performed by the robots. All these experiments have been carried out using simulated data. We assume that the robots are equipped with a stereo camera system, which enables them to obtain relative measurements to landmarks that are detected in the environment. In addition, we assume that a descriptor associated to the visual landmark can be computed from its visual appearance. The approach is valid for the case in which the robots obtain relative 3D measurements to a set of landmarks in the map. We also consider that the robots are able to communicate among themselves. In addition, we consider that the relative starting position of the robots is known in advance.

For this purpose, we have built an environment, shown in Figure 2. The simulated environment represents an office-like space of 40×40 meters in which there are several walls (the lines in Figure 2 represent the projection of these walls). Visual landmarks have been randomly distributed on these walls. These landmarks are represented by three global coordinates: (X_g, Y_g, Z_g) . Each landmark is characterized by a visual descriptor d_n obtained from real images of an indoor environment. For these experiments we use a map of 600 landmarks.

The experiments have been structured as follows: On the one hand, in Section IV-B we consider the situation in which all the robots start at the same point (marked as $P1$ in Figure 2). In this case, we consider that the relative initial position of the robots is known with respect to the global reference system. On the other hand, in Section IV-C the situation in which the robots start at different positions is also considered. In each situation we have compared the performance of the iFastSLAM algorithm when varying the number of robots and the number of particles in the filter. Finally, Section IV-D compares the computational cost of the presented approach with centralized algorithms that represent the joint probability of all the robots [9], [12].

A. Simulation details

The robots navigate within the described map and obtain relative measurements to 3D visual landmarks. At each time step each robot obtains a maximum number of observations B , each observation consists of a relative distance measurement $z_t^{(i)}$ and a descriptor $d_t^{(i)}$. In our case, we simulate that the robots

are equipped with a stereo camera placed at a fixed height and orientation. In consequence, we also test whether the landmark lies inside the field of view of the robot. We simulate the existence of walls that limit the visibility in the environment. For example, a landmark placed behind a wall will not be detected by the robot. Once we have performed this test, the observation will be formed by the relative position of the landmark referred to the robot reference system $z_t^{(i)} = (X_r, Y_r, Z_r)^T + \epsilon$, where $\epsilon = (\epsilon_X, \epsilon_Y, \epsilon_Z)^T$ is a gaussian noise vector distributed according to $N(0, R_t)$. In our case, we are obtaining measurements on point-like landmarks using a stereo vision system. The computation of the noise matrix R_t is described in [9].

At each time step, each robot sends its position, uncertainty and observations to the rest of the team members. Next, each robot integrates its own observations in its map and also the observations received from the rest of the team to compute its own global map.

B. Robots starting at the same initial point

In these experiments the robots start the mapping tasks at the same time and at the same location (point P1 in Figure 2) and the relative initial position between the robots is known. In these experiments we have 5 robots that perform different trajectories. According to the points marked in Figure 2, the trajectories followed by the robots are presented in Table I. Given these trajectories, we have varied the number of particles M in the filter and computed the RMS error in the estimated trajectory. The results are shown in Figure 3(a). In this figure, a comparison of the results varying the number of robots is performed. The algorithm has been tested using a particle filter with $M = \{1, 50, 100, 300, 500\}$ particles and $B = 7$ measurements for each robot. In Figure 3(a) we can observe a decreasing tendency in the error as the number of particles is higher, since the probability densities can be represented more precisely when more particles are used. It can also be observed that accurate results are obtained using $M = 100$ particles, whereas the results do not improve significantly with more than $M = 300$ particles. In addition, Figure 3(a) shows that the error in the estimated path decreases as the number of robots increases. This fact can be explained in the following way: when more robots are used in the SLAM task, the uncertainty in the pose has lower values, since each robot can localize itself when viewing landmarks previously seen by other robots. In addition, the map is estimated more precisely, since the number of observations grows with the number of robots in the team. Figure 3(a) presents the RMS error in the path when the experiment is repeated a number of times, showing the mean value and 2σ error bars.

TABLE I
TRAJECTORIES FOLLOWED BY EACH ROBOT

	Same starting position	Different starting position
Robot 1 (R1)	P1-P2-P3-P4-P5-P6-P1	P1-P2-P3-P4-P5-P6-P1
Robot 2 (R2)	P1-P6-P5-P4-P3-P2-P1	P3-P2-P1-P6-P5-P8-P9
Robot 3 (R3)	P1-P2-P3-P4-P9-P8-P7	P5-P4-P9-P8-P5-P4-P3
Robot 4 (R4)	P1-P6-P7-P8-P9-P4-P3	P7-P6-P1-P2-P3-P4-P5
Robot 5 (R5)	P1-P6-P5-P8-P9-P4-P3	P9-P4-P3-P2-P1-P6-P7

C. Robots starting at different initial points

Next, we present the same experiments for the case in which the robots start the exploration from different initial positions. Also the initial transformation between the robots is known. The trajectories followed by each robot are described in Table I. Paying attention to Figure 2, the robot R1 starts at point P1, R2 at P3, R3 at P5, R4 at P7 and R5 at P9. We repeated the experiments and also computed the RMS error in the path of robot R1 when varying the number of particles of the filter and present the results in Figure 3(b). Also, as in Figure 3(a) a clear decreasing tendency is observed when more particles are used in the estimation. The errors are also lower when more robots are used.

D. Computational cost

An iteration of the algorithm described here for a single robot with M particles that integrates B observations in a map with N landmarks requires a computational time of the form $O(MNB)$. If two robots construct the map independently, each one requires a computational effort of the form $O(M2 \cdot BN)$, since now, each robot must integrate its own observations and the observations received from the other robot. The total computational effort is thus $O(4MBN)$. In general, when K robots construct the map, the computational effort is of the form $O(K^2MBN)$, thus quadratic in the number of robots. [In order to compare with other solutions, please note that we are accounting for the total computational effort to construct the map. As mentioned before, in the proposed solution, this computational effort may be distributed across all the robots in the team, having each robot a computational cost of the form \$O\(MK \cdot BN\)\$, being \$K\$ the number of robots in the team.](#)

When the SLAM algorithm is based on Equation (1) (e.g. [9], [12], [20]) the number of particles needed to represent the state grows exponentially with the number of robots. If, for example, M particles are needed to represent the state (x, y, θ) of a single robot of dimension 3, M^2 particles will be needed to represent accurately the state of two robots, of dimension 6 and so on. Thus, since an iteration of the algorithm for a single robot that integrates B observations in a map with N landmarks requires a computational effort of the form $O(MNB)$. In consequence, an iteration of the joint algorithm with K robots requires $O(KM^KNB)$ in order to integrate KB measurements. Please note the exponential factor in the number of particles, which is avoided in the proposed algorithm.

Figure 4 presents a comparison of the total computational effort required to compute K independent maps using the independent FastSLAM algorithm (continuous line) and the time needed to compute a single joint map (dashed line). Being K the number of robots, we present the time needed to compute K independent filters with the proposed approach with the computation of a single unified filter. In order to compare we executed several times a simulation using the independent FastSLAM algorithm and also the joint FastSLAM algorithm. Due to practical reasons, the time required to finish each simulation varies slightly. In consequence, we present the mean value and standard deviation of different experiments. We present results using $M = 100$ particles, $B = 7$ measurements and varied the number of robots K . In Figure 4 it can be seen that the computational cost is of the form $O(K^2MBN)$ as the number of robots grows. In dashed line, we present the time needed to build a map using $M = 100$ particles (for a single robot) and $M = 100 \cdot 100$ particles for two robots. For clarity, we present results obtained using only 1 and 2 robots. Using M^K particles in the joint algorithm ensures that the probability over the path of every robot is represented with the same precision as the proposed approach.

In Figure 4 we can observe that the time required in the proposed approach is significantly lower than the joint FastSLAM approach. Although the computational cost is quadratic in the number of robots, we consider that the proposed algorithm is well suited to be computed independently on each robot. Thus, each robot in the team is able to compute a global map using its own onboard computer while sending his observations to other robots. In consequence, the total time to create the global map is of the form $O(KBNM)$, thus linear in the number of robots.

V. CONCLUSION

In this paper we present an approach that tackles the multirobot SLAM problem. Particularly, we focus on the situation in which a team of robots cooperate to build a feature-based visual map of the environment. In the presented approach, the relative initial location of the robots in the environment is

known. While travelling through the environment each robot obtains its own observations and integrates them in its own global map. In addition, each robot also incorporates the observations sent by other robots in his own map. Thus, all the robots in the team cooperate to compute several global maps. In this way, each robot in the team builds a different global map. The approach presented here differs from others in the sense that each robot maintains an independent SLAM filter, and still each robot can introduce the observations of other robots in his own map. In order to do this, in this paper we propose the use of a Rao-Blackwellized particle filter extended to the multi-robot case. The main difference with respect to previous approaches is the gaussian approximation over the path of other robots that allows to separate the SLAM problem into different particle sets. The measurements obtained by other robots are approximated using the mean value of the particles associated to its pose along with a linear error propagation.

Having several global maps may be beneficial in the context of multi-robot exploration, since the failure of a single team member will not affect the performance of the mission.

The results demonstrate that good results can be achieved using a reasonable number of particles. In addition, we show that the cooperation to build the map using several robots results in a better performance, since the robots can refine their location with respect to the observations introduced by other robots. We have also compared the computational effort required to build a map using a single unified FastSLAM filter with the time necessary to compute K independent filters. The results, show that the computational effort is significantly lower in the presented approach and is more scalable than the joint computation of the filter. In addition, we consider that having separate particle sets benefits the computation of the maps, since each robot may use its own onboard computer to build a global map, thus reducing the total time required.

APPENDIX: EQUATIONS

Next we would like to compute the local measurement that would be observed by robot $\langle i \rangle$, if it could sense the landmark seen by robot $\langle j \rangle$. That is, we would like to compute $\bar{z}_t^{(i,j)}$:

$$\bar{z}_t^{(i,j)} = H^{-1(i)} H^{(j)} z_t^{(j)} \quad (4)$$

where $\bar{z}_t^{(i,j)}$ is given in local coordinates of the reference system of robot $\langle i \rangle$. The matrix H is a homogeneous transformation matrix that computes the global coordinates of a landmark given a relative measurement z_t . Note that $H^{(j)}$ is computed using the pose of robot $\langle j \rangle$, whereas $H^{-1(i)}$ is calculated using the pose of robot $\langle i \rangle$. Equation (4) represents a transformation of the measurement obtained by

the robot $\langle j \rangle$ to global coordinates and, next, to the local coordinates of robot $\langle i \rangle$. Equation (4) can be written in detail as:

$$\bar{z}_t^{(i,j)} = H^{-1(i)} H^{(j)} z_t^{(j)} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 & -x \cos(\theta) - y \sin(\theta) \\ -\sin(\theta) & \cos(\theta) & 0 & x \sin(\theta) - y \cos(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{(i)} \begin{pmatrix} \cos(\bar{\theta}) & -\sin(\bar{\theta}) & 0 & \bar{x} \\ \sin(\bar{\theta}) & \cos(\bar{\theta}) & 0 & \bar{y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{(j)} \begin{pmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{pmatrix}^{(j)}$$

where the matrix $H^{(j)}$ is computed with the mean of the particles in the filter of the robot $\langle j \rangle$, assuming a gaussian distribution $\bar{x}_t^{(j)} = (\bar{x}, \bar{y}, \bar{\theta})^{(j)}$. On the other hand $H^{-1(i)}$ is computed independently with the pose $x_t^{[m](i)} = (x, y, \theta)^{(i)}$ of each particle in the filter $S_t^{(i)}$.

The integration of the measurements $\bar{z}_t^{(i,j)}$ coming from other robots can be obtained by means of the computation of an extended error matrix $R_e^{(i,j)}$ by means of the following equation:

$$R_e^{(i,j)} = J R_t^{(j)} J^T + Q F_u^{(j)} Q^T \quad (5)$$

where $J = \nabla_{z_t^{(j)}} z_t^{(i,j)}$ and $Q = \nabla_{x_t^{(j)}} \bar{z}_t^{(i,j)}$ are Jacobian matrices that linearly approximate Equation (5). The matrix $R_t^{(j)}$ is the noise matrix associated with robot's $\langle j \rangle$ onboard sensor. On the other hand, the matrix $F_u^{(j)}$ is the covariance matrix computed from the particle cloud associated to robot $\langle j \rangle$ at time t . In this case, the uncertainty of the particles associated to robot $\langle j \rangle$ is approximated by a gaussian distribution. The Jacobians J and Q can be computed as:

$$J = \begin{pmatrix} \cos(\theta^{(i)} - \theta^{(j)}) & \sin(\theta^{(i)} - \theta^{(j)}) & 0 \\ -\sin(\theta^{(i)} - \theta^{(j)}) & \cos(\theta^{(i)} - \theta^{(j)}) & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad (6)$$

$$Q = \begin{pmatrix} \cos(\theta^{(i)}) & \sin(\theta^{(i)}) & 0 & X_r \sin(\theta^{(i)} - \theta^{(j)}) - Y_r \cos(\theta^{(i)} - \theta^{(j)}) \\ -\sin(\theta^{(i)}) & \cos(\theta^{(i)}) & 0 & X_r \cos(\theta^{(i)} - \theta^{(j)}) - Y_r \sin(\theta^{(i)} - \theta^{(j)}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}^{(j)} \quad (7)$$

REFERENCES

- [1] M. Ballesta, A. Gil, O. Reinoso, and L. Payá. Building visual maps with a team of mobile robots. *Ed. IN-TECH. Multi-Robot Systems, Trends and Development*, pages 95–108, 2011.
- [2] M. Ballesta, A. Gil, O. Reinoso, L. Pay, and L. M. Jimenez. Map fusion in an independent multi-robot approach. *WSEAS TRANSACTIONS on SYSTEMS*, 9(9):959–968, 2010.
- [3] M. Ballesta, O. Reinoso, A. Gil, Payá L., and M. Juliá. Map fusion of visual landmark-based maps. In *Proc. of the 40th International Symposium on Robotics*, pages 75–81, Barcelona, Spain, 11-13 March, 2009.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, pages 511–517, 2006.

- [5] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. Moon, and J.-B. Song. Safe navigation of a mobile robot considering visibility of environment. *IEEE Trans. on Industrial Electronics*, 56(3):3941–3950, 2009.
- [6] E. Cuevas, F. Wario, D. Zaldivar, and M. Pérez-Cisneros. Circle detection on images using learning automata. *IET Comput. Vis.*, 6(2):121–132, 2012.
- [7] R. Garcia-Garcia, M.A. Sotelo, I. Parra, D. Fernandez, Naranjo J.E., and M. Gavilan. 3D visual odometry for road vehicles. *Journal of Intelligent and Robotic Systems*, 51(1):113–134, 2008.
- [8] A. Gil, Ó. Martínez-Mozos, M. Ballesta, and Ó. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Machine Vision and Applications*, pages 905–920, 2011.
- [9] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá. Multi-robot visual SLAM using a Rao-Blackwellized particle filter. *Robotics and Autonomous Systems Journal*, pages 68–80, 2010.
- [10] K. Gurkahraman, E. Unsal, and Y. Cebi. Omni-directional vision system with fibre grating device for obstacle detection. *IET Comput. Vis.*, 5(5):267–281, 2011.
- [11] M. Tan Han and Y. F. Li. Ceiling-based visual positioning for an indoor mobile robot with monocular vision. *IEEE Transac. on Industrial Electronics*, 56(5):1617–1628, 2009.
- [12] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *The International Journal of Robotics Research*, 25:1243–1256, 2006.
- [13] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map merging for distributed robot navigation. In *Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 212–217, 2003.
- [14] N. Kwak, G-W. Kim, S-H. Ji, and B-H. Lee. A mobile robot exploration strategy with low cost sonar and tungsten-halonen structural light. *Journal of Intelligent and Robotic Systems*, 51(1):89–111, 2008.
- [15] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4):376–382, 1991.
- [16] D. Lowe. Object recognition from local scale-invariant features. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, pages 1150–1157, Kerkyra, Greece, 1999.
- [17] A. Mancini, A. Cesetti, A. Iuale, E. Frontoni, P. Zingaretti, and S. Longui. A framework for simulation and testing of UAVs in cooperative scenarios. *Journal of Intelligent and Robotic Systems*, 54(1-3):307–329, 2009.
- [18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In *Eighteenth national conference on Artificial Intelligence*, pages 593–598, Menlo Park, CA, USA, 2002.
- [19] A. C. Murillo, J. J. Guerrero, and C. Sagüés. SURF features for efficient robot localization with omnidirectional images. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3901–3907, San Diego, CA, USA, 2007.
- [20] J. Nieto, J. Guivant, E. Nebot, and S. Thrun. Real time data association for FastSLAM. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 412–418, 2003.
- [21] R. Orghidan, E.M. Mouaddib, J. Salvi, , and J.J. Serrano. Catadioptric single-shot rangefinder for textured map building in robot navigation. *IET Comput. Vis.*, 1(2):43–53, 2007.
- [22] S. Palaniswamy, N.A. Thacker, and C.P. Klingenberg. Automatic identification of landmarks in digital images. *IET Comput. Vis.*, 4(4):247–260, 2010.
- [23] L. Parker. Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems*, 14(6):3–12, 2000.

- [24] A. Ramisa, A. Tapus, D. Aldavert, R. Toledo, and R. Lopez de Mantaras. Robust vision-based robot localization using combinations of local feature region detectors. *Journal of Autonomous Robots*, 27:373–385, 2009.
- [25] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *Int. Journal of Robotics Research*, 20(5):335–363, 2001.
- [26] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005. ISBN: 0-262-20162-3.
- [27] R. Triebel and W. Burgard. Improving simultaneous mapping and localization. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 545–551, 2005.
- [28] J. Valls-Miró, W. Zhou, and G. Dissanayake. Towards vision based navigation in large indoor environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2096–2102, Beijing, China, 2006.
- [29] O. Wijk and H. I. Christensen. Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 1(31):31–42, 2000.
- [30] X.Z. Zhang, A.B. Rad, Y.K. Wong, G.Q. Huang, Y.L. Ip, and K.M. Chow. A comparative study of three mapping methodologies. *Journal of Intelligent and Robotic Systems*, 49(4):385–395, 2007.
- [31] Xun S. Zhou and Sergios I. Roumeliotis. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, pp. 1785-1792, 2006.

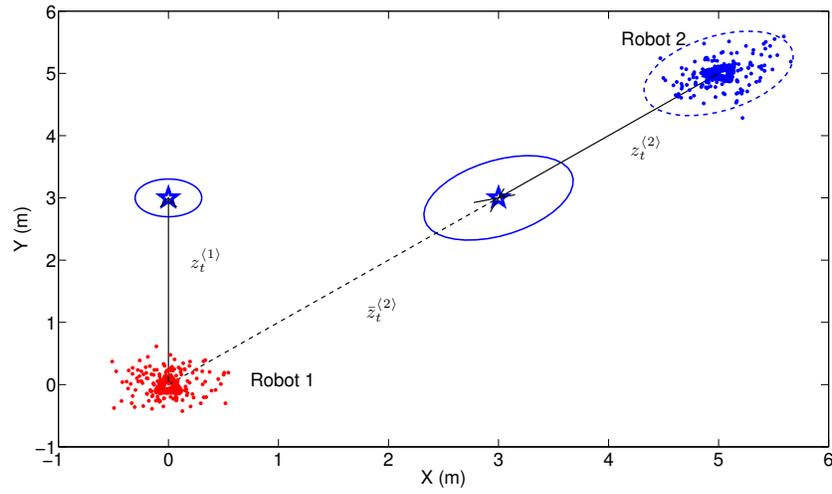


Fig. 1. The figure presents two robots: Robot 1 and Robot 2, marked with triangles. Two landmarks are marked with stars. In this figure, the uncertainty of Robot 1 is represented by a set of particles. In the case of Robot 2, the set of particles is approximated by a gaussian distribution (dashed ellipse). In the figure, Robot 1 integrates an own observation $z_t^{(1)}$ and, in addition, it also integrates the observation of Robot 2.

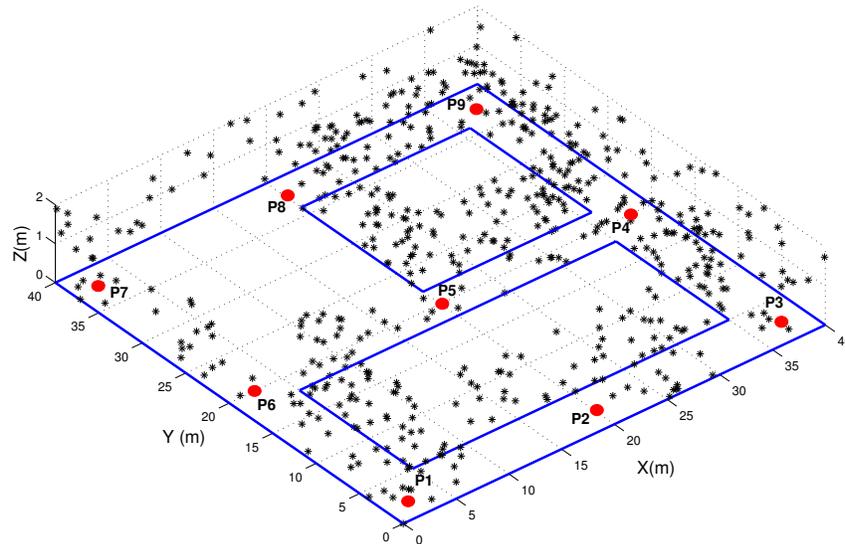
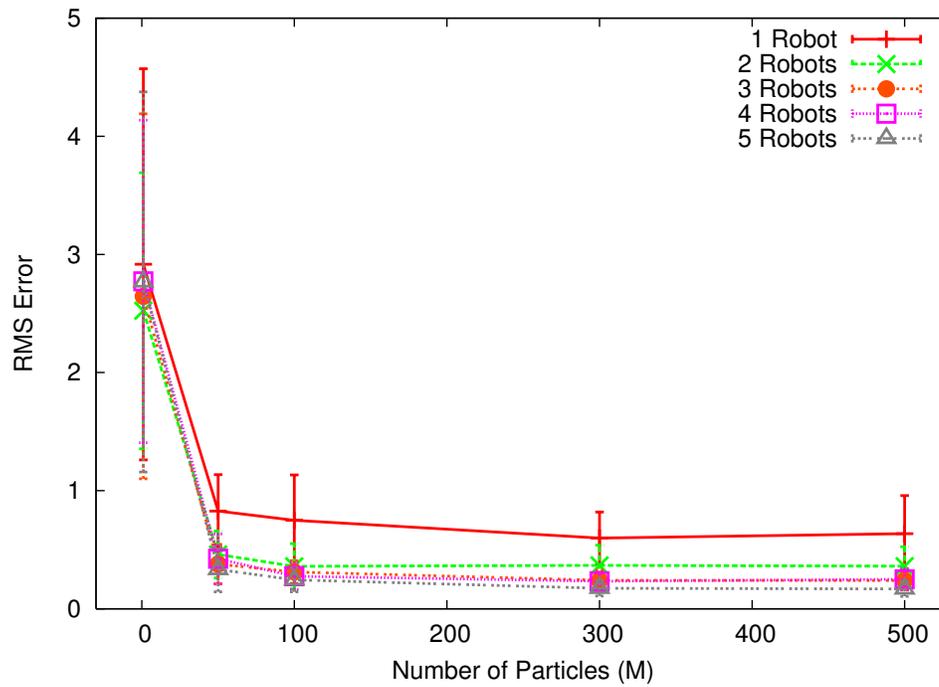
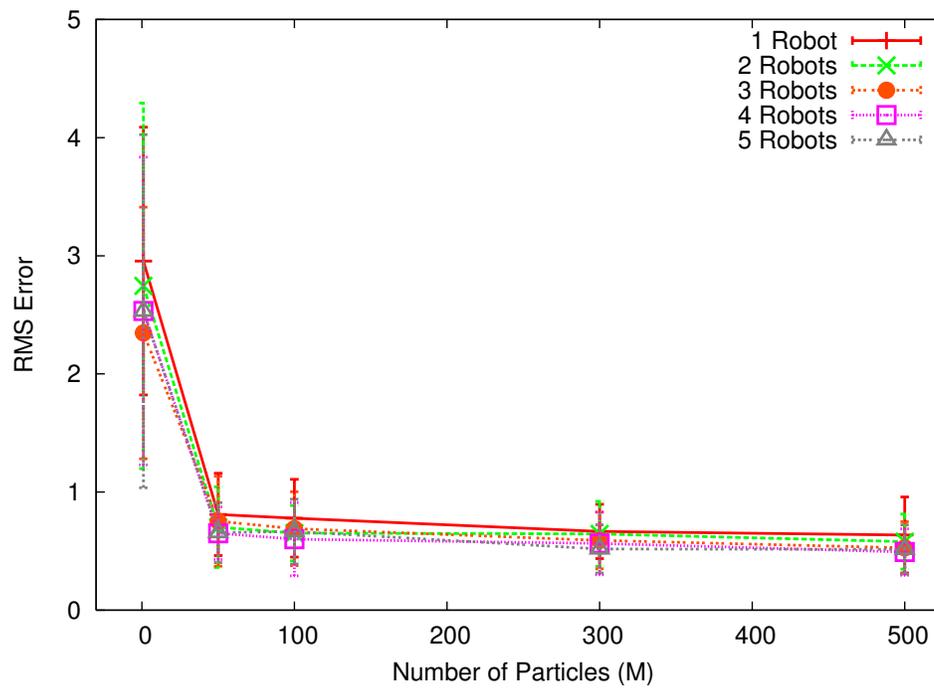


Fig. 2. The figure presents a 3D view of the simulated environment.



(a)



(b)

Fig. 3. RMS Error vs. Number of particles (M). 3(a) robots starting from the same location. 3(b) robots starting from different positions.

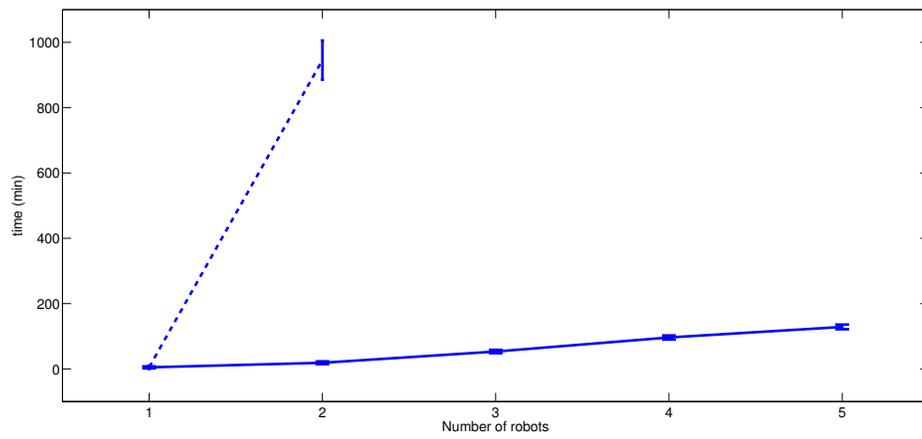


Fig. 4. The figure presents the total computational time of the proposed approach (continuous line) and the joint probability approach (dashed line).