Multiple Robots in a Cooperative Task: Exploration and Mapping.

Adão de Melo Neto, Paulo Fernando Ferreira Rosa, Thiago Eustaquio Alves de Oliveira, Paulo César Pellanda

***Abstract*— In this work is investigated the exploration of an environment with multiple vehicles using a strategy based on occupancy grids and a technique of simultaneous localization and mapping (SLAM). The exploration strategy uses concepts of costs and utility from frontier-cells. Besides, the used SLAM method is based on a FastSLAM algorithm with landmarks extracted from visual sensors and a features map common to vehicles. Both activities - location of the vehicles and exploration of the environment - are coordinated by a central agent. The results show that when two vehicles can communicate with a central agent building a features map common to vehicles, the exploration task becomes more efficient than that performed with dedicated maps, because the accuracy of vehicle position and orientation is increased with the use of an even number of particles. In this paper we also present and evaluate the implementation of the approach in a real environment.**

I. Introduction

Integrated exploration of an environment is a high-level activity where methods of exploration, locating, mapping and navigation have to be combined so that autonomous vehicles be able to map the environment maximizing gains (e.g., accuracy of position and orientation - pose - of the vehicles) and minimizing costs (e.g., time spent in the exploration).

SLAM is a robotic research field related to the vehicle ability to locate itself and obtain a features map of landmarks (something that can be easily detected and described) of the environment. Localization and mapping are interdependent and related functionalities since if, on the one hand, it is necessary to know the vehicle location to build a map of the environment in which it is situated, on the other, an environment map is needed to locate the vehicle. In many cases, for vehicle location, is used the combination of a GPS (global positioning system) with an IMU (inertial measurement unit). However, the information quality obtained through this system may be affected by the number of satellites in view and electromagnetic interference as well as by odometry errors. As result, they have a low precision for navigation in an indoor environment. The SLAM problem can be stated as follows [1]: given an autonomous vehicle within an unknown environment and, using only observations relative to detectable landmarks in the environment in relation to the vehicle, build a features map for those landmarks and simultaneously compute an estimation of the vehicles location based on this map. In this work is used a SLAM technique known as FastSLAM [2] to locate the vehicles. Since a SLAM algorithm is ensuring the location of a vehicle, an occupancy grid can be simultaneously generated to assist the environment exploration [9], considering that the workspace has a limited size.

This research is carried out under a grant from FAPERJ (Foundation for Research of the State of Rio de Janeiro).

Defense Engineering Graduate Program Military Institute of Engineering, Rio de Janeiro, Brazil, Tel: 55-21-25467092, E-mails: adao@ime.eb.br, rpaulo@ime.eb.br, thiago.eustakio@gmail.com, pellanda@ime.eb.br.

The objective of this paper is to show that we can explore an indoor environment with multiply vehicles in a efficient way when we are using a FastSLAM algorithm so as to locate them, considering landmarks extracted through of SIFT algorithm [13], a computer vision algorithm used to detect and describe features in images, and a features map common to vehicles. In this approach, the vehicles increase the accuracy in the pose in relation the exploration with the use of dedicated maps. The improvement in accuracy, due to the use of a common features map, is obtained using the same number of particles (a particle represents an estimate of the pose of the vehicle and of the features map of landmarks detected - Eq.5). The main contribution of this paper is to show that the approach increases the accuracy in the pose of the vehicles, as well as present and evaluate the implementation of the approach using vehicles pioneer 3DX with sensors LMS-200 and LMS kinect [15]. Section II discusses some works related to SLAM. We treat FastSLAM considering one and two vehicles in sections III and IV. Section V focuses on exploration strategy, and the section VI describe experiments and an evaluation of the approach. We explain our conclusions in the section VII.

II. Related Works

Several studies on mapping have already been published. Some approaches estimate the vehicles pose only using odometers, which often leads to inaccuracy. Others use SLAM techniques (EKF-SLAM [1] and FastSLAM [2]) with extraction of features from raw data provided by laser and sonar sensors ([3]). In [4] a Rao-Blackwellized particle filter [2] is applied to estimate simultaneously the map and the path of a single vehicle. In the mentioned work, SIFT features are used as landmarks in the environment and extracted using a pair of stereo cameras. SLAM approaches with multiple vehicles can be grouped into two solutions. In the first group, each vehicle estimates its own individual map using its observations and, at a later stage, a common map is formed by fusing the individual maps of the vehicles. In the second one, the estimation of the trajectories and the map are made jointly. A single map is computed simultaneously by using the observations of all the vehicles. The work presented in [6] and [8] can be classified in the first group. In [6], each robot builds its own map and at the same time continuously attempts to localize in the maps built by

other vehicles using particle filters. The approach can cope with the situation where the initial locations of the vehicles are unknown, however, the fusion of the individual maps is computationally expensive. [8] proposes an algorithm for multiple vehicles based on [2] where they can start from unknown pose in advance. In this approach, each vehicle builds and maintains its own map. When they are on the same line of sight, the maps are fused. The approach presented in [7] and [5] belong to the second group. [7] uses an extended Kalman filter (EKF) to estimate a state vector formed by the poses of all the vehicles and a set of 2D landmarks. The vehicles obtain observations and construct a single unified map using the update equations of the classical EKF [1]. The initial positions of the vehicles must be known in advance and the data association is assumed to be known. In this case, the main drawback stems from the fact that a single hypothesis about the pose of vehicle is maintained. [5] presents an algorithm based on [2] where the map is common for vehicles and it is assumed they have initial pose known in advance. The authors show, through experiments conducted in the *Matlab* and the observation of visual landmarks, that the approach proposed is suitable for a small groups of vehicles. As regards the exploration strategies, we highlight the approaches described in [10] that uses concepts of costs and utility from frontier-cells.

This work shows that in the context of an integrated exploration with multiply vehicles (we used two vehicles for validation), that the FastSLAM approach with a common map increases the accuracy in the pose of vehicles. The FastSLAM approach differs from EKF-SLAM for multiple hypothesis data association. Four experiments were conducted. In the first a *Player/Stage* simulator was used to test the complete approach, considering each 3D visual landmark represented by an image that has only a SIFT feature. In the second one, the data association is analyzed using images taken with a *kinect* visual sensor, an active stereo sensor which reduces the correspondence problem in a stereo pair and has a effective range of 0.8 to 3.5 meters. In the third and fourth experiments a Pioneer 3DX vehicle with laser and *kinect* sensors is used. The main contribution of this paper is to show that the above approach increases the accuracy in the pose of the vehicles presenting and evaluating the implementation of the approach using a Pioneer 3DX vehicle with sensors laser and *kinect*.

III. FastSLAM

FastSLAM [2] decomposes the SLAM problem into a vehicle localization problem, and a collection of landmark estimation problems that are conditioned to the vehicle pose estimation. Let u_t be a control action responsible for the exchange of state of a vehicle at time t. In robotics, the pose χ_t of a vehicle and the observation z_t of a landmark θ_j are modeled by probabilistic laws

$$p(\chi_t|\chi_{t-1}, u_t) \text{ and } p(z_t|\chi_t, \theta_j) \tag{1}$$

with values sampled by functions usually nonlinear in their arguments (h and g) with a Gaussian noise added with mean 0 and, respectively, covariance Q_t and R_t:

$$\chi_t = h(u_t, \chi_{t-1}) + N(0, Q_t) \text{ and } z_t = g(\chi_t, \theta_j) + N(0, R_t) \tag{2}$$

FastSLAM estimates the posterior probability distribution of the vehicle's path $\chi^t = \{\chi_1, \ldots, \chi_t\}$ and the map Θ (Eq.3) considering the observations $z^t = \{z_1, \ldots, z_t\}$, control actions $u^t = \{u_1, \ldots, u_t\}$ and associations $a^t = \{a_1, \ldots, a_t\}$ between the features of landmark that were observed and the features of landmarks in the map

$$p(\chi^t, \Theta|z^t, u^t, a^t) \tag{3}$$

It is shown that if the path χ^t is known, then the position of landmarks θ_i in Θ are conditionally independent, which allows to factor the problem of estimating the posterior probability distribution of χ^t and Θ as a product of simple terms (Eq.4).

$$p(\chi^t, \Theta|z^t, u^t, a^t) = \underbrace{p(\chi^t|z^t, u^t, a^t)}_{Path} \prod_{n=1}^{N} \underbrace{p(\theta_n|\chi^t, z^t, u^t, a^t)}_{Landmark} \tag{4}$$

Posterior probability distribution of χ^t is estimated using a particle filter [11] and the posterior probability distribution for the N landmarks θ_i of each particle are estimated by N Extended Kalman filters (EKF) [11] conditioned to the path χ^t. The particle filter represents the distribution using a set $S_t = \{S_t^{[1]}, \ldots, S_t^{[M]}\}$ of particles

$$S_t^{[m]} = \left[\chi_t^{[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, ..., \psi_{n,t}^{[m]}, \Sigma_{n,t}^{[m]}, d_n^{[m]}, ...}_{\Theta^{[m]}}, w_t^{[m]} \right] \tag{5}$$

where, $\psi_{n,t}^{[m]}$ and $\Sigma_{n,t}^{[m]}$ are the mean value and covariance for the coordinates of θ_n conditioned to the path $\chi^{t,[m]}$; $d_n^{[m]}$ is the descriptor of its features and $w_t^{[m]}$ is the weight of the particle. When the vehicle makes an observation it must update its map. Since the sensors are prone to errors, each information embedded in the map may have a certain amount of uncertainty. This inaccuracy can lead to errors in data association and in update. Update errors can be minimized through successive observations and, therefore, in order to improve accuracy in the map, it is necessary to *close the loop*, i.e., observe again a landmark previously observed. This error is also due to the linearization performed by EKF. Error in data association, caused by the erroneous association (correspondence) of features of a landmark on map and features of a landmark observed can be avoided using a robust set of features, as is the case of SIFT features, whose descriptor, a vector of dimension 128 is invariant to scale and rotating of the image and partially invariant to changes in lighting and 3D viewpoint of the camera. The FastSLAM algorithm used here is shown in sequence. We consider an observation $v_t = \{z_t, d_t\}$ where d_t is the SIFT descriptor of a visual landmark and $z_t = [d_x, d_y, d_z]^T$ the distance between the landmark and the vehicle (Fig.1).

FastSLAM (S_{t-1}, z_t, d_t, u_t)

1. $S_t = S_{aux} = \emptyset$
2. For each particle $S_{t-1}^{[m]}$ in S_{t-1}
   1. Sample $\chi_t^{[m]} \sim p(\chi_t | \chi_{t-1}^{[m]}, u_t)$
   2. For each landmark $\theta_n^{[m]}$ in map $\Theta^{[m]}$ compute
      1. $E_n^{[m]} = Mahalanobis(d_n^{[m]}, d_t) = (d_n^{[m]} - d_t)(d_n^{[m]} - d_t)^T$
      2. $p_{n,t}^{[m]} = f(z_t, \theta_n^{[m]})$
   3. End For
   4. j = Find $p_{n,t}^{[m]} \geq P_0$
   5. $a_t = argmin_j \; E_j^{[m]}$
   6. If $E_{a_t}^{[m]} \leq E_0$
      1. Update of landmark $\theta_{a_t}^{[m]}$
      2. $w_t^{[m]} = p_{a_t,t}^{[m]}$
   7. Else
      1. New landmark in $\Theta^{[m]}$
   8. End If
   9. $S_t^{[m]} \Rightarrow S_{aux}$
3. End For
4. id =Best_particle (S_t)
5. Resampling(S_t)
6. Return S_t

For each particle $S_{t-1}^{[m]}$ in S_{t-1}, the vehicle movement model is sampled (step 2.1, Eq.1, Eq.2). On the other hand, for each landmark $\theta_n^{[m]}$ in the map $\Theta^{[m]}$ is computed the distance $E_n^{[m]}$ between the SIFT descriptor d_t of the landmark observed and the descriptor $d_n^{[m]}$ of $\theta_n^{[m]}$ (step 2.2.1), and the quality of the association of $\theta_n^{[m]}$ with z_t through a function of the FastSLAM (step 2.2.2). The data

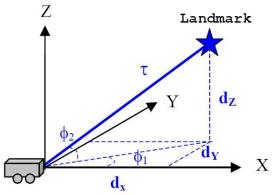


Fig. 1. Observation Model of a visual landmark 3D.

association chooses a set j of landmarks with association probability greater than P_0 (step 2.4) and select, in this set, the landmark that minimizes $E_j^{[m]}$ (step 2.5). If $E_{a_t}^{[m]}$ is less than a threshold value E_0 (step 2.6), the data association is considered correct and the estimation of the coordinates of landmark $\theta_{a_t}^{[m]}$ is updated with equations of the EKF (step 2.6.1). Otherwise, a new landmark is created on the map (step 2.7.1)[5]. The weight of the particle w_t corresponds to quality of the association $p_{t,a_t}^{[m]}$ of landmark $\theta_{a_t}^{[m]}$ which is associated with the observation z_t (step 2.6.2). The best particle - which corresponds to a estimate of location - has the greatest weight (step 4). In resampling step (step 5), particles with higher weight $w_t^{[i]}$ are replicated.

The FastSLAM algorithm in the case of one vehicle requires time MN. This is because M particles need to be processed, whereas, for each particle the data association needs to iterate over the N landmarks in the map. However, if each particle is stored in a *kd-tree* structure with dimension of the SIFT descriptor (128), the research in each structure by a list of landmarks (nearest neighbors at a distance E_0) costs $logN$, which accelerates the data association, implying a time of $MlogN$ (step 2.2). In this list, is chosen those which have association probability greater than P_0 (step 2.4).

The success obtained with SLAM has motivated the research on SLAM with multiple vehicles, as discussed in the next section.

IV. FastSLAM with Multiple Vehicles

When multiple vehicles have the possibility to communicate with a central coordinator agent, they can work together to reduce the exploration time and allows a cooperation in observing landmarks from the environment.

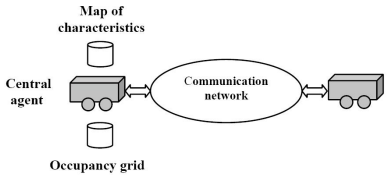


Fig. 2. Scheme of exploration with two vehicle in which a central agent is responsible for building the map (localization) and the grid (exploration).

If k vehicles explore the environment building a common map (Fig.2), and at instant t the vehicle (i), in pose $\chi_{t,(i)}$, performs a single observation $z_{t,(i)}$, the posterior probability distribution of the path $\chi_{(1:k)}$ of k vehicles and the map Θ can be estimated from the following function [5]:

$$p(\chi_{(1:k)}^t, \Theta | z_{(1:k)}^t, u_{(1:k)}^t, a^t) = \underbrace{p(\chi_{(1:k)}^t | z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{Path} \prod_{n=1}^{N} \underbrace{p(\theta_n | \chi_{(1:k)}^t, z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{Landmark}$$

where $\chi_{(1:k)}^t = \{\chi_{(1)}^t, \ldots, \chi_{(k)}^t\}$, $u_{(1:k)}^t = \{u_{(1)}^t, \ldots, u_{(k)}^t\}$ and $z_{(1:k)}^t = \{z_{(1)}^t, \ldots, z_{(k)}^t\}$ are respectively the set of paths, actions, and observation of k vehicles and $a^t = \{a_1, \ldots, a_t\}$ is the data associations history. Posterior probability distribution of $\chi_{(1:k)}^t$ is estimated using k particles filters. On the other hand, the posterior probability distribution of N landmarks θ_i, corresponding to each particle, is estimated by kN independent extended Kalman filters, conditional on the paths $\chi_{(1:k)}^t$. Since the map is common to vehicles, k particle filters produce the same set S_t of particles

$$S_t^{[m]} = \left[\chi_{(1:k),t}^{[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, \ldots, \psi_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}, d_N^{[m]}}_{\Theta^{[m]}}, w_t^{[m]} \right] \quad (6)$$

Note that the state to be estimated is composed by the pose $\chi_{t,(1:k)}$ of the k vehicles. An algorithm for multiple vehicle SLAM is shown as follows [5]. The routine FastSLAM* in algorithm corresponds to the FastSLAM algorithm of the section III considering only steps 1-3.

FastSLAM with Multiple (two) Vehicles

1. $S_t = \emptyset$
2. For $t = 1$ to End do
   1. $[z_{t,(1)}, d_{t,(1)}, z_{t,(2)}, d_{t,(2)}]$ = Observations()

2.$[S_t, w_{t,(1)}] = FastSLAM^*(S_{t-1}, z_{t,(1)}, d_{t,(1)}u_{t,(1)})$
3.$[S_t, w_{t,(2)}] = FastSLAM^*(S_{t-1}, z_{t,(2)}, d_{t,(2)}u_{t,(2)})$
4.$w_t = w_{t,(1)}w_{t,(2)}$
5.$S_t =$ Resampling (S_t, w_t)
3. End For

This algorithm consider that the vehicles begin exploration from a pose known by the central agent. Considering that vehicles start from nearby positions, the pose relative can be obtained by laser sensor. Since the map is common to vehicles, for each particle $S_t^{[m]}$ defined in equation 6, k weights are calculated and the total weight associated with particle $S_t^{[m]}$ is defined as

$$w_t = \prod_{i=1}^{k} w_{t,(i)}^{[m]} \tag{7}$$

FastSLAM algorithm in the case of k vehicles require time $kMlogN$.

We present below the adopted exploration strategy.

V. Exploration Strategy

In an exploration activity, the path to be followed by a vehicle must be controlled for the sake of efficiency. In the considered approach [10], a target - frontier-cell of an occupancy grid [9] - is chosen according to a function that evaluates the cost of navigation and the utility of the target.

A. Costs

In order to determine the cost of reaching the current frontier-cells, we compute the optimal path from the current position of the vehicle to all the frontier-cells based on a deterministic variant of the *value iteration*, a popular dynamic programming algorithm [14]. It is considered that the cost for traversing a grid cell (x, y) is proportional to its occupancy value $P(occ_{xy})$. The minimum-cost path is computed using the two steps below:

1) **Initialization.** The grid cell that contains the location of vehicle is initialized with 0, and all others with ∞

$$V_{x,y} \leftarrow \begin{cases} 0, & \text{if (x,y) is the vehicle position} \\ \infty, & \text{otherwise} \end{cases}$$

2) **Update loop.** For all grid cells (x, y) do

$$\begin{array}{c} V_{x,y} \leftarrow \\ min\{V_{x+\Delta x,y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} P(occ_{x+\Delta x,y+\Delta y})\} \\ | \quad \Delta x, \Delta y \in \{-1, 0, 1\}, P(occ_{x+\Delta x,y+\Delta y}) \in [0, occ_{max}] \end{array}$$

where occ_{max} is the maximum occupancy probability value of a grid cell the vehicle is allowed to transverse. This technique updates the value of all grid cells by the value of their best neighbors, plus the cost of moving to this neighbor. Here, cost is equivalent to the probability $P(occ_{x,y})$ that a grid cell (x, y) is occupied times the distance to the cell. The update rule is repeated until convergence. All in all each value $V_{x,y}$ corresponds to the cumulative cost of moving from the current position of the vehicle to (x, y).

B. Utilities of frontier-cells

If there is already a vehicle that moves to a frontier-cell, the utility of this cell must be lower for other vehicles. Let us suppose that in the beginning each frontier-cell t has the utility U_t which is equal for all frontier-cells. Then, we compute the utility $U(t_n|t_1, \ldots, t_{n-1})$ of a frontier-cell t_n given that the cells $t_1, \ldots, t_{n-1}$ have already been assigned to the vehicles $1, \ldots, n-1$ as

$$U(t_n|t_1, \ldots, t_{n-1}) = U_{t_n} - \sum_{i=1}^{n-1} P(\underbrace{||(t_n - t_i||}_{d}) \tag{8}$$

where $P(d)$ is the probability that the sensor of vehicle (a laser sensor in our case). will cover cells in distance d. According to Equation 8, the more vehicles move to a location from where t_n is likely to be visible, the lower is the utility of t_n. We compute $P(d)$ as

$$P(d) \leftarrow \begin{cases} 1 - \frac{d}{max\_range}, & if \quad d < max\_range \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where max_range is the maximum range reading provided by sensor.

C. Target Point Selection

In the selection of destinations, is considered for each vehicle i, a balance between the cost V_t^i of moving to a destination t and the utility U_t of this target [10].

The next section describes the experimental set up and some results.

VI. Experiments and Results

Four experiments were conducted. In the first, a *Player/Stage* simulator (Fig. 3) was used to test the complete approach. In the second one, only the data association is verified through images obtained with a *kinect* sensor, an active stereo sensor, in a pre-defined path (Fig. 7) of a real environment. In the third and fourth experiments a pioneer 3DX vehicle with laser LMS-200 and *kinect* sensors is used. The results of the first and fourth experiments were obtained considering the use of dedicated maps (case "a") and common map (case "b").

A. Experiment 1

In the first experiment, a laser sensor Sick LMS-200 embedded in each Pioneer 3DX were utilized to build the occupancy grid. On the other hand, to simulate a stereo visual sensor with a field of view of 180^o, 3D landmarks represented by images were used. Each landmark contains only one SIFT feature and its coordinates $[\theta_x, \theta_y, \theta_z]^T$ were artificially assigned (Fig.1). The images are differents and therefore, the data association is known. When a visual landmark is within the field of vision, the SIFT algorithm extracts the descriptor of image and compute the distance τ and the orientations ϕ_1 and ϕ_2 of landmark in relation to the vehicle (Fig.1).

Selection of destinations was done as described in section V, where an occupancy grid with cells of size 0.5m x 0.5m

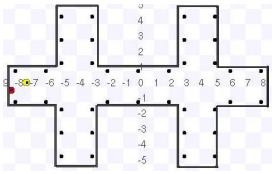


Fig. 3. Explored environment (17m x 10.5m) with vehicles in their initial pose

TABLE I

SAMPLING OF THE MOTION (FIG.4) AND OBSERVATION (FIG.1) MODEL

$N(0,\sigma)$ is a normal distribution with zero mean and standard deviation σ. Values in meters and radians.

Variable	Average error
$x_{t+1} = x_t + \Delta x_t + \overline{d}N(0,\sigma)$	0 ($\Delta x_t = 0$) or $\overline{d} = \frac{5}{100}d$ ($\Delta x_t \neq 0$)
$y_{t+1} = y_t + \Delta y_t + \overline{d}N(0,\sigma)$	0 ($\Delta y_t = 0$) or $\overline{d} = \frac{5}{100}d$ ($\Delta y_t \neq 0$)
$\varphi_{t+1} = \varphi_t + \Delta\varphi_t + \overline{g}N(0,\sigma)$	$\overline{g} = \frac{3\pi}{180}$
$\tau = \tau + \overline{\tau}N(0,\sigma)$	$\overline{\tau} = 0.1$
$\phi_i = \phi_i + \overline{\phi_i}N(0,\sigma)$	$\overline{\phi_i}$

was used. After selecting the destination, the path to be traveled by the vehicle was computed using *A-star* algorithm [12]. The position of the vehicle (x and y) and its orientation (φ) at each step in the occupancy grid (Fig. 4), distance τ and orientations ϕ_1 and ϕ_2 (Fig.1) were sampled according to the table I.

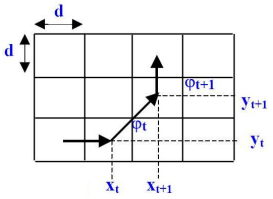


Fig. 4. Motion model used

In the experiment, the following parameters were adopted: $\sigma = 0.5$, $\overline{\phi_i} = \frac{0.1\pi}{180}$ rad, $d = 0.5$ m, $E_0 = 0$, $P_0 = 0.9$, 2000 particles and range of the visual sensor of 3.5 meters. Since each landmark has only one (unique) SIFT descriptor, we did $E_0 = 0$. Our goal is to show that in the context of a perfect data association, the approach with common map increases the accuracy in the pose of the vehicles. Figure 5 and 6 show the trajectories traveled by the vehicles in tasks of exploration.

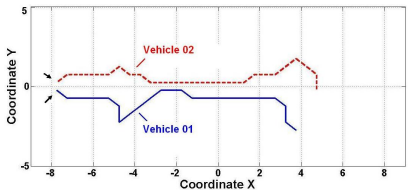


Fig. 5. Trajectory traveled by vehicles in an exploration task: vehicle 01 (continuous line) and vehicle 02 (dashed line). The arrows indicate the initial position of vehicles. Total number of steps per vehicle: 28.

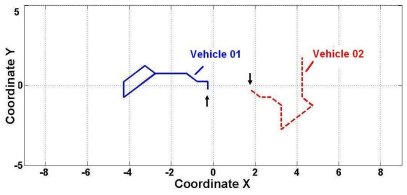


Fig. 6. Trajectory traveled by vehicles in an exploration task: vehicle 01 (continuous line) and vehicle 02 (dashed line). The arrows indicate the initial position of vehicles. Total number of steps per vehicle: 16.

The results of the table II is related to figure 5. The results of the tables III is related to figure 6. In table II we can see that the accuracy and amount of estimates of location obtained in exploration of the case "b" (common map) increased in relation to the case "a" (dedicated maps). Note that we are using a very restrictive condition P_0 to ensure a good accuracy.

TABLE II

RESULTS IN PLAYER/STAGE SIMULATOR (FIG.5)

Error (RMS) in coordinates x,y and orientation α; amount of estimate of location obtained divided by the number of steps used in exploration; and amount of updates of landmark on the map.

Case	x	y	α	$\frac{Estimate}{Step}$	Update
a 1	0.042	0.040	**0.108**	**16/28**	**106**
a 2	0.042	0.037	**0.105**	**17/28**	**71**
b 1	0.045	0.043	**0.066**	**23/28**	**180**
b 2	0.038	0.041	**0.073**	**25/28**	**215**

Maximum and minimum values (average $\pm$ standard deviation)

Case	x_{min}	x_{max}	y_{min}	y_{max}	α_{min}	α_{max}	$\frac{Estimate}{Step}$
a 1	0.020	0.064	0.022	0.058	**0.066**	**0.150**	**57%**
a 2	0.019	0.064	0.022	0.052	**0.065**	**0.145**	**61%**
b 1	0.022	0.068	0.031	0.055	**0.037**	**0.096**	**82%**
b 2	0.025	0.050	0.029	0.053	**0.052**	**0.095**	**89%**

TABLE III

RESULTS IN PLAYER/STAGE SIMULATOR (FIG.6)

Error (RMS) in coordinates x,y and orientation α; amount of estimate of location obtained divided by the number of steps used in exploration; and amount of updates of landmark on the map

Case	x	y	α	$\frac{Estimates}{Steps}$	Update
a 1	0.027	0.032	0.062	13/16	92
a 2	0.032	0.029	0.066	13/16	82
b 1	0.026	0.032	0.069	14/16	140
b 2	0.029	0.028	0.059	15/16	143

Maximum and minimum values (average $\pm$ standard deviation)

Case	x_{min}	x_{max}	y_{min}	y_{max}	α_{min}	α_{max}	$\frac{Estimate}{Step}$
a 1	0.018	0.037	0.021	0.042	0.032	0.093	81%
a 2	0.015	0.048	0.019	0.039	0.034	0.097	88%
b 1	0.018	0.034	0.022	0.042	0.037	0.101	87%
b 2	0.017	0.043	0.022	0.034	0.035	0.084	93%

When a vehicle is exploring an environment, the accuracy of the estimated pose tends to decrease, since it depends on the estimated accuracy of the landmarks mapped. In turn, the estimated coordinates of the landmarks mapped depend

on the accuracy of the estimated pose of the vehicle, which tends to decrease as the vehicle explores the environment if it does not occur corrections through observations of landmarks mapped previously. Therefore, a greater number of updates of landmarks tends to improve the accuracy of the pose estimated by SLAM and it is better that they occur as often as possible. The SLAM allows to estimate the pose of the vehicle based on the estimation error of the odometer of the vehicle and of the model of observation. This estimate of the pose is used to correct the error in odometry of the vehicle, considering that the environment is small, since the estimate depends on the accuracy of the landmarks detected. The use of FastSLAM with common map is justified by the fact that the update of landmarks tends to happen more often, and in a shorter distance traveled by the vehicle. Even with a increase in the size of the problem to be estimated by FastSLAM (pose of two vehicles), the experiments confirms that this approach produces a good estimate of vehicle location, because a vehicle can update a landmark detected by itself or by another vehicle, so that, probabilistically, updates occur more often. Using a common map is more advantageous in the situation where a vehicle has to pass close to a region where another vehicle has already covered (Fig.5). In this approach, although the choose of best particle is made individually by each vehicle, the weight of each particle is the product of the weight of the estimates of each vehicle (Eq.7) and the resampling of particles is done based on this weight, which means that a particle updated only by a vehicle will be resampled to a less proportion. However, the SIFT algorithm allows to obtain many landmarks and the addition of more vehicles will increase the number of updates of landmarks, since this update can be done in a landmark mapped by any of the vehicles. The results of table III confirm that the FastSLAM with common map is a good estimator because when a vehicle does not update landmarks on map inserted by another vehicle (the sensor range is 3.5 meters and the path of the vehicles are different), the results (accuracy and number of estimates of location by step) are similar to FastSLAM with dedicated maps.

B. Experiment 2

In the third experiment, a path (Fig.7) in an environment containing chairs, desks, computers, so on, was travelled with a visual sensor *kinect* and, at each step, an image and a file containing the distances $[d_x, d_y, d_z]^T$ of pixels to the sensor was obtained and stored to then be used to simulate a vehicle running a FastSLAM algorithm.

The figure 8 shows the variation in number of updates on the map, considering, respectively, the variation in limits P_0 and E_0 (Fig. 8). In the simulation, the distance τ and the orientations ϕ_1 and ϕ_2 (Fig.1) for SIFT landmarks detected were obtained from the distance $[d_x, d_y, d_z]^T$. The following parameters were adopted: 1000 particles, $\sigma = 0.5$, $\overline{\phi_i} = \frac{0.1\pi}{180}$ rad, $d = 1.0$ m and effective range of *kinect* (0.8 to 3.5 meters). We can conclude that, respectively, a P_0 too high and a E_0 too low may inhibit updates. On the other hand, a value of P_0 much low with a value of E_0 much high can degrade the map and the FastSLAM algorithm. The ideal is a high value of P_0 and a value of E_0 sufficiently low.

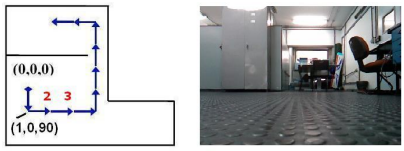


Fig. 7. Path (10 meters) in an unstructured environment containing desks, chairs, computers, and so on, traveled - of simulated manner - by the vehicle (left) and picture of the environment mentioned (right)

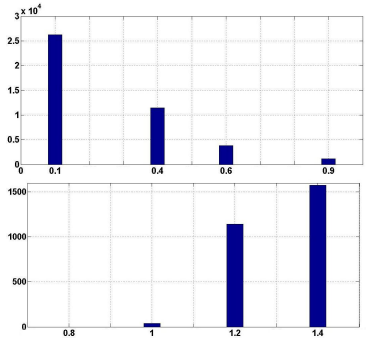


Fig. 8. Average number of updating on map in function of P_0 ($E_0 = 1.2$) - above - and E_0 ($P_0 = 0.9$) - below.

To define the value of E_0, we did a separate analysis of SIFT descriptors regardless of the effective range of the *kinect*. We obtained the descriptors of all images (3125)and compute, for each descriptor, the Mahanalobis distance to other descriptors. We found that the descriptors obtained in a same image are different, ie, all distances are different from 0. The average Mahanalobis distance was equal to 1.2. The table IV shows the percentage of distances between the descriptors of all images that are below certain limits.

TABLE IV
ANALYSIS OF DESCRIPTORS 2

Distance-Limit (E_0)	Quantity of distances	Percentage
0.1	0	0
0.3	252	0.003
0.5	6362	0.065
0.7	53588	0.550
0.9	298898	3.061
1.1	1559184	15.971
1.2	3319498	34.002

As the SIFT descriptor is partially invariant to the point of sight 3D and lighting, we did an analysis of the correspondence (association) of the descriptors of two images (Fig.9). From the results obtained and consolidated in the table V, we found that the values $E_0 = 0.4$ and $E_0 = 0.5$ provides a good combination in relation to the number of correct correspondences (large) and to number of correspondences possible (little). In that table the calculations were made considering that there are SIFT features corresponding in the two images (Fig.9). However, in the FastSLAM the correspondence (association) is made between an observed

landmark of an image and the landmarks of the feature map of a particle. Let's assume that the landmark observed does not have a correspondence on the map (is the first time what was observed), then, an association (erroneous) will occur only if the position of the associated landmark is very close to the predicted position by the particle of the FastSLAM, since we are using $P_0 = 0.9$ (a high value). Therefore, the lower the value of E_0, the better the quality of the data association, since we have a smaller number of possible correspondences (Tab.V).

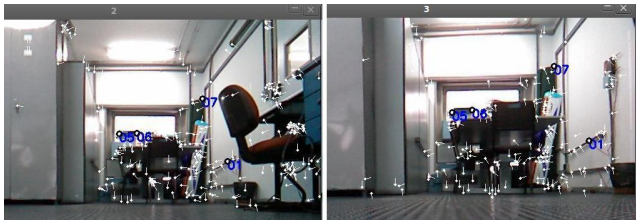


Fig. 9. Correspondence of descriptors of the images obtained in steps 2 and 3 (Fig.7) considering $E_0 = 0.4$

TABLE V
ANALYSIS OF DESCRIPTORS 3

E_0	$\frac{correct\_associations}{Total\_associations}$
0.4	$04/04 = 100\%$
0.5	$13/14 = 93\%$
0.6	$22/25 = 88\%$
0.7	$31/42 = 74\%$

C. Experiment 3

In the fourth experiment we traveled an environment (Fig.10) with a Pionner 3DX vehicle following the path defined in the figure 10. In the experiment, the following parameters were adopted: $\sigma = 0.5$, $\overline{\phi_i} = \frac{0.1\pi}{180}$ rad, 1000 particles and effective range of *kinect*.

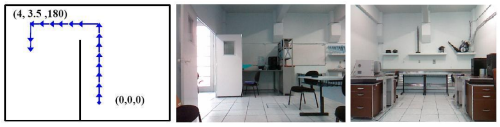


Fig. 10. Path (16 steps) travelled by Pioneer 3DX (Each step has 0.5 or 1.0 meters) and corresponding images in the poses (0,0,0) and (4,3.5,180).

In the figure 11 we have the processing time per step in function on number of landmarks on the map. The total time spent was 494 sec. Assuming that each step we expect 10 seconds for the displacement of the vehicle, the total processing time of the FastSLAM was 334 seconds (5.56 min or 21 sec/steps). If we had not used the *kd-tree* structure, the time would grow linearly with the number of landmarks on the map. In 50% of steps was obtained an estimate of location. In the figure 12 we have the higher resolution grid - *laser map* - considering in each step the best particle.

In order to assess the impact of the choice of E_0 in processing time of the FastSLAM, from the images and from

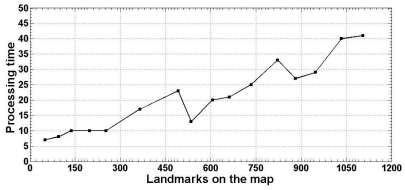


Fig. 11. Processing time per step in function on number of landmarks on the map ($P_0 = 0.9$ and $E_0 = 0.5$)

Fig. 12. Higher resolution grid - *laser map* - generated by best particle ($P_0 = 0.9$ and $E_0 = 0.5$).

the file containing the distances $[d_x, d_y, d_z]^T$ of pixels in relation to sensor that was stored during the experiment, we get the amount of estimates of location, amount of updates of landmarks and processing time (Tab.VI) considering 1000 particles. From the results we can conclude that the processing time not varied considerably.

TABLE VI
RESULTS IN FUNCTION OF E_0 CONSIDERING 1000 PARTICLES.

E_0	$\frac{Estimates}{Steps}$	Updates	Time (sec)	Time (min)
0.3	3/16	03	299	4.98
0.4	6/16	15	319	5.31
0.5	8/16	37	334	5.56
0.6	9/16	88	369	6.15

D. Experiment 4

In the fifth experiment, from the images and files containing the distances $[d_x, d_y, d_z]^T$ of the pixels in relation to *kinect* of the experiment 4, we simulate two vehicles traveling the paths described in figure 13 and using FastSLAM with dedicated maps (case "a") and common map (case "b").The following parameters were adopted: 1000 particles, $\sigma = 0.5$, $\overline{\phi_i} = \frac{0.1\pi}{180}$ rad and effective range of the *kinect*. This experiment considered in case "b" that a

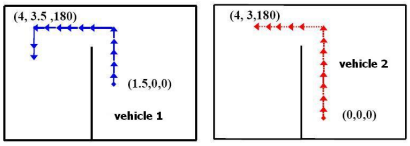


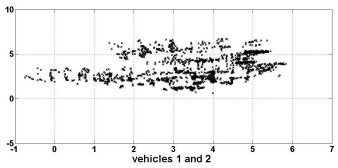
Fig. 13. Path (13 meters) traveled - of simulated manner - by vehicle 01 (continuous line) and 02 (dashed line). Each step has 0.5 or 1.0 meters.

vehicle will only insert a new landmark on the map if the region was not mapped by another vehicle (the vehicle 2 will insert landmarks only in the step 1 and 2). In this case the vehicle 2 (case "b") obtained a greater amount of estimate of location and updates of landmarks (Tab.VII). This

implies that there was observations of landmarks previously observed by vehicle 1. The figure 14 shows the landmarks map generated by the best particle. In the case "b", each particle which updated a landmark, did it in just a few of them ($P_0 > 0.9$), $N_{cm} \simeq 0.57 * (N_{dm_1} + N_{dm_2})$ and the searching of nearest neighbors selected 1.98 times more landmarks.

TABLE VII
RESULTS CONSIDERING IN FUNCTION OF $E_0 = 0.5$ AND CONSIDERING 1000 PARTICLES.

Case	vehicle	$\frac{Estimates}{Steps}$	Updates	Time (min)
a	1	05/13	38	4.10
a	2	08/13	47	3.71
b	1	05/13	54	9.98
b	2	12/13	831	9.98



Case "b" - Common map

Fig. 14. 2D Map generated by the best particle ($P_0 = 0.9$, $E_0 = 0.5$)

E. Evaluation of the approach

The results show that using the same number of particles, the exploration conducted with common map increases the accuracy in pose of vehicles. It was expected that the increase in number of vehicles used in the exploration entailed the need to increase the number of particles in order to achieve a similar pose error estimation, which turned out to be unnecessary. In both approaches, the accuracy achieved by each vehicle depends on the updates of landmarks. When using a common map to vehicles, this number tends to increase because these updates can be made also in a landmark previously mapped by another vehicle. When using a common map, all processing will get concentrated in the central agent, which should have a capacity equivalent to the number of vehicles used (two in our case) so that time spent in the exploration is similar. The approach with common map requires a time $2MlogN_{cm}$ instead of $MlogN_{dm}$ ($N_{cm} \simeq 0.5*(N_{dm_1}+N_{dm_2})$). Although there is an increased computational cost, the order of complexity remains $O(MlogN_{cm})$ and landmarks are stored in a *kd-tree* structure. Therefore, this increase can be compensated by a greater computational power in central agent or paralleling the approach with common map. The processing of landmarks observed by vehicles can be parallelized, since a particle represents an estimate of the pose of the vehicle with their respective set of estimated landmarks. It is expected that the processing time with common map and dedicated maps are approximately identical. In relation to complexity in extraction of landmarks with algorithm SIFT, we consider that it is done individually by each vehicle. Regarding the need for communication for the transmission of descriptors for the central agent, this communication is also required for transmission of upgrades of the occupation grid.

VII. CONCLUSION

In this work we show that we can explore an indoor environment with multiply vehicles in a efficient way when we are using a FastSLAM algorithm so as to locate them, considering landmarks extracted through of SIFT algorithm and features map common to vehicles. This is because the vehicles increases the accuracy in the pose, everything in relation the use of dedicated maps. The improvement is obtained with the same number of particles, which could potentially be larger, since the estimated problem is greater.

REFERENCES

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics And Automation*, Vol. 17, Issue 3, pp. 229-241.

[2] M. Montemerlo and S. Thrun (2003). Simultaneous localization and mapping with unknown data association. *Proceedings of the IEEE, International Conference on Robotics and Automation*, Vol. 2, pp. 1985-1991.

[3] C. Fulgenzi, G. Ippoliti, S. Longhi (2009). Experimental validation of FastSLAM algorithm integrated with a linear features based map. *Elsevier: Mechatronics*, Vol. 19, Issue 5, pp. 609-616

[4] A. Gil, O. Reinoso, O. Martnez-Mozos, C. Stachniss, W. Burgard (2006). Improving Data Association in Vision-based SLAM. *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp. 2076-2081

[5] A. Gil, O. Reinoso, M. Ballesta, M. Julia (2009). Multi-robot visual SLAM using Rao-blackwellized particle filter. *ACM: Robotics and Autonomous Systems*, Vol. 58, Issue 1, pp. 68-80.

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

[7] J.W. Fenwick, P.M. Newman, J.J. Leonard (2002). Cooperative concurrent mapping and localization. *IEEE Int. Conf. on Robotics and Automation, ICRA*, pp. 1810-1817

[8] A. Howard (2006). Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, Vol. 25, Issue 12, pp. 1243-1256.

[9] A. A. Makarenko, S. B. Willians, F. Bourgault, H. F. Durrant-Whyte (2002). An Experiment of Integrated Exploration. *Proceedings of the 2002 IEEE-RSJ. Intl Conference on Intelligent Robots and Systems*, Vol. 1, pp. 534-539.

[10] W. Burgard, M. Moors, C. Stachniss and F. Schneider (2005). Coordinated Multi-Robot Exploration, *IEEE Transactions on Robotics, Vol. 21, Issue 3, pp. 376-386*.

[11] S. Thrun, W. Burgard and D. Fox (2005). Probabilistic robotics. *Massachusetts Institute of Technology Press*, Cambridge, MA, USA, 2005.

[12] Hart, P. E., Nilsson, N. J., Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, Vol. 4, Issue 2, pp. 100107

[13] David Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 2, pp. 91-110

[14] R. Howard (1960). Dynamic Programming and Markov Process. *MIT Press and Wiley*.

[15] Stowers, J. and Hayes, M. and Bainbridge-Smith, A. (2011). Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor. *IEEE International Conference on Mechatronics (ICM)*. pp. 358 -362