

A FastSLAM-based Algorithm for Omnidirectional Cameras

Cristina Gamallo, Manuel Mucientes and Carlos V. Regueiro

Abstract—Environments with a low density of landmarks are difficult for vision-based Simultaneous Localization and Mapping (SLAM) algorithms. The use of omnidirectional cameras, which have a wide field of view, is specially interesting in these environments as several landmarks are usually detected in each image. A typical example of this kind of situation happens in indoor environments when the lights placed on the ceiling are the landmarks. The use of omnivision combined with this type of landmarks presents two challenges: the data association and the initialization of the landmarks with a bearing-only sensor. In this paper we present a SLAM algorithm based on the well-known FastSLAM approach [1]. The proposal includes a novel hierarchical data association method based on the Hungarian algorithm, and a delayed initialization of the landmarks. The approach has been tested on a real environment with a *Pioneer 3-DX* robot.

Index Terms—Simultaneous Localization and Mapping; FastSLAM; Omnidirectional camera; Hungarian Association;

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) has attracted the attention of many researchers in the last decade. The first proposals were based on Extended Kalman Filters (EKF) using range sensors like laser range scanners [2], [3] or ultrasound sensors [4], [5]. The use of cameras for SLAM, known as visual SLAM, is a more recent line of research and has received an increasing attention in the last years. The first work for visual SLAM was presented by Davison in 1998 [6]. The use of cameras is interesting as they are low-cost, light and compact sensors and, also, because they provide richer information of the environment, as colour and texture.

One of the most widely used configurations in visual SLAM has been stereo vision [7], [8], [9]. Stereo cameras can provide 3D information into a single measurement, so traditional SLAM algorithms can be applied without modifications. The main drawback of this type of sensor is the limited 3D range. On the other hand, the use of monocular cameras allows the detection of very far objects. However, they are bearing-only sensors, i.e., they do not provide information about distance or depth. Therefore, a mechanism to estimate the 3D position of the landmarks needs to be incorporated.

An important issue in visual SLAM is data association, which solves the correspondence between measurements and

landmarks. The most simple technique for data association is Maximum Likelihood (ML) [10], which assigns the measurement to the nearest landmark that has not been associated yet. This approach is the most widely used for SLAM [11], [7], [12]. However, this technique is brittle if there are several equally likely hypotheses (Fig. 1(a)). This ambiguity is usual in robotics when landmarks are indistinguishable among each other, and it comes from two sources: i) pose uncertainty (Fig. 1(b)) and ii) because landmarks can be quite close to each other and have uncertainty.

If a wrong data association is picked, this decision can have a catastrophic result on the accuracy of the resulting map. Several strategies [13] have been developed in order to deal with ambiguity in noisy environments. For example, Monte Carlo Data Association [14] assigns the correspondences probabilistically in accordance to their likelihoods. However, as happens with ML, it is a local method and it considers each measurement independently to establish the correspondences, and not the set of measurements as global methods do.

Global methods make the joint data association between all measurements and landmarks based on a global score. This type of methods are better to cope with the ambiguity introduced by the noise of the sensors and the pose uncertainty. A typical representative of global methods is the Hungarian algorithm [15], which solves linear assignment problems in polynomial time for several measurements and landmarks. Another approach is presented in [16], [5], where they process multiple observations jointly, and consider the geometric relationships between a set of landmarks to test the correspondence vector. This method can only be applied on approximations that use features of the environment with geometric relationships.

In this paper, we present a SLAM algorithm for omnivision. The omnidirectional camera has a fish-eye lens with a very wide field of view (FOV). The landmarks of the environment are the ceiling lights and, therefore, the camera is equipped with a band-pass IR filter. This makes features extraction easier, as the filter only detects those objects that emit in the IR spectrum. The proposal is based on the FastSLAM [1] algorithm, but modified for bearing-only sensors. The main contributions of the paper are: i) the data association, which is hierarchical and based on the Hungarian algorithm [15], to deal with the bearing-only sensor and, also, because the landmarks are indistinguishable among each other; ii) the initialization of the landmarks, that takes into account a bunch of measurements associated to the candidate landmark.

The paper is structured as follows. First, we briefly review the related work for bearing only visual SLAM. Sec. III

Cristina Gamallo and Manuel Mucientes are with the Centro de Investigación en Tecnologías de la Información (CITIUS), Universidad de Santiago de Compostela (Spain). E-mail: cristina.gamallo@usc.es, manuel.mucientes@usc.es

Carlos V. Regueiro is with the Department of Electronics and Systems, University of A Coruña (Spain). E-mail: cvazquez@udc.es

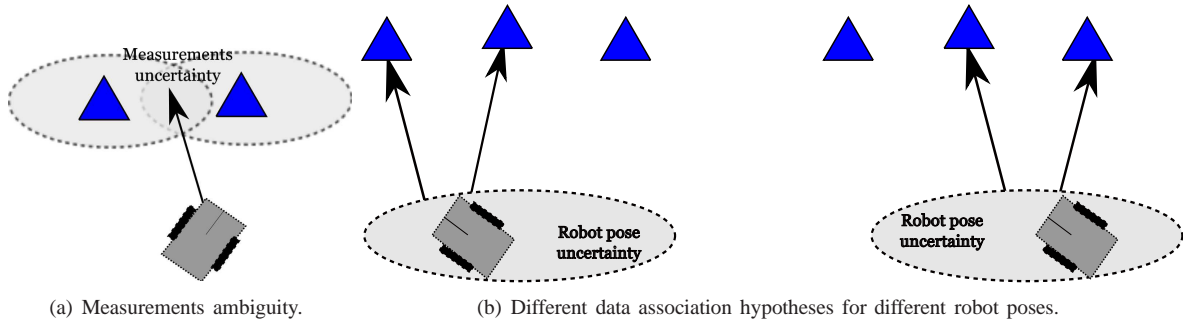


Fig. 1. Typical problems for data association in SLAM. Landmarks are represented by triangles, and the measurements by arrows.

describes the proposed algorithm and Sec. IV presents the experimental results. Finally, Sec. V points out the conclusions.

II. RELATED WORK IN BEARING-ONLY VISUAL SLAM

Monocular cameras are bearing-only sensors, as they only provide information on the orientation of the objects detected in the image. As depth information is not available, 3D positions of objects cannot be obtained with a single image. The approaches to solve this problem, known as landmark initialization in bearing-only SLAM, can be grouped in two categories: delayed and undelayed solutions [17]. A representative approach to undelayed initialization is [18]. The initial state of the landmarks was approximated with a sum of Gaussians that augment the state of the EKF. However, the most usual approach is the delayed initialization mechanism, which requires the use of several images to estimate the 3D positioning of the landmark. [19] presents a landmark initialization algorithm based also in a sum of Gaussians but without including this information in the state vector. Also, in [20] a particle filter to estimate the initial position of the landmarks was described.

The use of omnivision cameras in SLAM allows to track the detected features over long distances, as these cameras have a very wide FOV. Therefore, the initialization process is well conditioned by the numerous observations of the same landmarks. The first paper, as far as we know, that used this type of camera was [21]. In that proposal, the landmarks were initialized using a delayed mode for two poses: the triangulation of the measurements was compared to existing relationships of the tags already in the map. In [22], the true location of the landmarks was approximated by the intersection point of two lines. Finally, [11] presents a minimalist approach based on a topological map for environments of medium to large size.

Although most of the visual SLAM approaches are based on EKFs [20], [22], [23], there are also approaches based on FastSLAM [12], on decoupling the pose error from the map error [21], etc.

III. SLAM ALGORITHM

The solution to the SLAM problem presented in this work is an extension of the FastSLAM 2.0 algorithm [1]. The novelties of the proposal are the data association process, and the initialization of the landmarks. This modifications are

necessary as the sensor is bearing-only, and the landmarks are indistinguishable among each other. FastSLAM algorithms use a Rao-Blackwellized particle filter [10], i.e., a filter that represents the posterior with a combination of particles and Gaussians. For FastSLAM, the particles estimate the robot path, while the landmarks are filtered with EKFs. The main contributions of our approach are:

- **Hierarchical data association based on the Hungarian algorithm.** As the sensor is bearing-only, and the landmarks are indistinguishable, it is necessary to implement a data association method. Our algorithm classifies the landmarks in two categories: regular landmarks (named landmarks in what follows) and candidate landmarks. The landmarks compose the map, while the candidate landmarks are those for which initialization was not possible yet. The data association has to take into account that the candidate landmarks are not reliable, i.e., most of the candidate landmarks will disappear, and only few of them will be transformed to landmarks. Therefore, priority is given to the association of the measurements to the current landmarks, while only those measurements that were not associated in the first stage will be included for the association with the candidate landmarks. This hierarchical data association is based on the Hungarian algorithm for each of the levels of the hierarchy. The Hungarian method is able to obtain the best association between the set of measurements and the set of landmarks in polynomial time.
- **Landmarks initialization.** We propose an initialization mechanism for the landmarks in which the 3D position of each landmark is obtained through several consecutive detections from different positions. Moreover, the process requires to approximate the inverse model of the camera with a look-up table, in order to obtain the angles of each feature from the pixels in the image.

Our proposal is shown in Alg. 1. It receives the set of N_{z_t} measurements (z_t) at the current time t , the control (u_t) and the previous set of particles (Y_{t-1}). Each particle k contains an estimated robot pose denoted as (x_{t-1}^k), a map of N_{t-1}^k landmarks $\{B_{1,t-1}^k, \dots, B_{N_{t-1,t-1}^k}^k\}$, and the set of η_{t-1}^k candidate landmarks $\{C_{1,t-1}^k, \dots, C_{\eta_{t-1,t-1}^k}^k\}$. A landmark j is defined with a Gaussian of mean $\mu_{j,t-1}^k$ and covariance $\Sigma_{j,t-1}^k$, and with the number of times it has been detected ($i_{j,t-1}^k$). Each candidate landmark has a set of measurements

Algorithm 1 FasSLAM algorithm for Omnivision.

SLAM (Z_t, u_t, Y_{t-1})

- 1: **for** $k = 1$ to M **do**
- 2: Get particle k from Y_{t-1} :

$$x_{t-1}^k, \{B_{1,t-1}^k, \dots, B_{N_{t-1,t-1}^k}\},$$

$$\{C_{1,t-1}^k, \dots, C_{\eta_{t-1,t-1}^k}\}$$
- 3: $\hat{x}_t = g(x_{t-1}^k, u_t)$
- 4: *measurementsLikelihood*()
- 5: $\Psi = \text{dataAssociation}(\Phi)$
- 6: *robotPoseUpdate*()
- 7: *landmarksUpdate*()
- 8: **end for**
- 9: $Y_t = \text{sampling}(\hat{Y}_t)$

$\{z_{t-n}^k, \dots, z_{t-1}^k\}$ that could be transformed into a landmark if the initialization conditions are fulfilled. The steps of the algorithm can be grouped in the following blocks:

- A main loop that iterates for each of the M particles (Alg. 1 lines 1- 8) to obtain the particle weights (w^k).
 - **Measurements likelihood.** For each combination of landmarks and measurements, calculate the likelihood of the association ($\phi_{l,j}$) (Alg.2).
 - **Data association.** Solve the data association (Alg. 1 line 5).
 - **Robot pose update.** The mean and covariance of the proposal distribution will be calculated with the contribution of each of the associated landmarks. Finally, the pose of the robot will be sampled from that distribution (Alg. 3).
 - **Landmarks update.** Update each landmark using an EKF and calculate the importance weight (w^k) for each particle (Alg. 4).
- **Resampling.** The new set of particles (Y_t) is generated by sampling the updated particle set (\hat{Y}_t) with probabilities proportional to the particle weights (w^k) using low variance sampling (Alg. 1 line 9).

In the following subsections, the measurement model, the data association, the robot pose update, the landmarks update, and the landmarks initialization will be described in more detail.

A. Measurement model

The sensor model is a feature-based model, where the features are the lights placed on the ceiling of the environment. These features are extracted from the images obtained by an omnidirectional camera following a detection process that was widely described in [24]. The output of the feature extraction process is a list of pixel coordinates (u_l, v_l) , that represent the centroid of each feature l . This list must be transformed into a measurements list $(z_{t,l})$, where each measurement is given by the azimuth and elevation angles $(\varphi_{l,t}, \theta_{l,t})$.

The camera follows a projection model developed by Pajdla and Bakstein [25] that indicates how a 3D point can be transformed to a pixel in a 2D image. The model is described

as a function of two angles, the azimuth (φ) and the elevation (θ):

$$\left. \begin{aligned} r &= a * \tan \frac{\theta}{b} + c * \sin \frac{\theta}{d} \\ u_l &= u_0 + r * \cos \varphi \\ v_l &= \beta * (v_0 + r * \sin \varphi) \end{aligned} \right\} \quad (1)$$

where a, b, c, d are parameters of the model, (u_0, v_0) are the coordinates of the pixel at the center of the image, and β is the ratio between the width and the height of a pixel.

The transformation of the measurements requires the inverse camera model, i.e., given a pixel the inverse model returns the coordinates of the 3D point in the world. However, the camera model equations are not invertible. This has been solved through a look-up table: given the coordinates of a pixel, the look-up table provides the values of φ and θ . Fig. 2 shows a graphical representation of the look-up table. The table only needs to be generated once, and this can be done off-line. The process is as follows:

- 1) Sample the values of φ and θ with precisions δ_φ and δ_θ . Equations 1 are used to obtain the corresponding pixel coordinates.
- 2) Store, for each pixel, the maximum and minimum values of φ and θ , as a range of values could correspond to the same pixel.

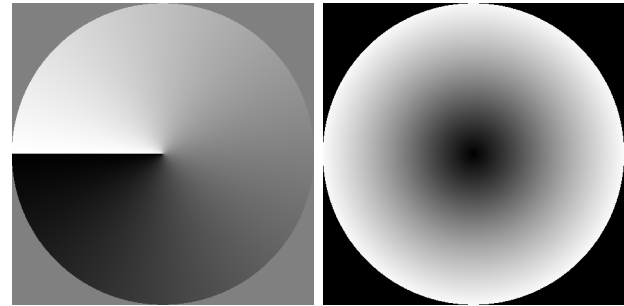


Fig. 2. Graphical representation of the values of φ and θ provided by the look-up table for each image pixel.

B. Measurements likelihood

The set of data associations (Ψ) between the measurements and the landmarks is decided based on the probability that feature l corresponds to landmark j ($\phi_{l,j}$) (Alg. 2). The loop from lines 1 to 13 iterates for all landmarks in order to estimate all the $\phi_{l,j}$ values (Alg. 2 line 11). Each probability depends on the measurement $(z_{t,l})$, the predicted measurement $(\hat{z}_{l,j})$, Alg. 2 line 10) and the measurement innovation covariance matrix $(Q_{j,l})$ (Alg. 2 line 6).

The measurement innovation covariance matrix $(Q_{j,l})$ is calculated taking into account the noise in the measurement (Q_t) , the previous covariance of the landmark $(\Sigma_{j,t-1}^k)$, and the Jacobian of h (measurement model) with respect to the measurement model variables. Moreover, the predicted covariance of the robot pose taking into account landmark j

Algorithm 2 Measurements likelihood algorithm

MeasurementsLikelihood()

- 1: **for** $j = 1$ to N_{t-1}^k **do**
- 2: $\bar{z}_j = h(\mu_{j,t-1}^k, \hat{x}_t)$
- 3: $H_{x,j} = \nabla_{x_t} h(\mu_{j,t-1}^k, \hat{x}_t)$
- 4: $H_{m,j} = \nabla_{m_j} h(\mu_{j,t-1}^k, \hat{x}_t)$
- 5: **for** $l = 1$ to N_{z_t} **do**
- 6: $Q_{j,l} = Q_{l,t} + H_{m,j} \Sigma_{j,t-1}^k H_{m,j}^T$
- 7: $\Sigma_{x,j,l} = [H_{x,j}^T Q_{j,l}^{-1} H_{x,j} + R_t^{-1}]^{-1}$
- 8:
- 9: $\mu_{x_t,j,l} = \Sigma_{x,j,l} H_{x,j}^T Q_{j,l}^{-1} (z_{t,l} - \bar{z}_j) + \hat{x}_t$
- 10: $\hat{z}_{l,j} = h(\mu_{j,t-1}^k, \mu_{x_t,j,l})$
- 11:
- 12: **end for**
- 13: **end for**

$$\phi_{l,j} = (2\pi)^{-\frac{Dim(Q_{j,l})}{2}} |Q_{j,l}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (z_{t,l} - \hat{z}_{l,j})^T Q_{j,l}^{-1} (z_{t,l} - \hat{z}_{l,j}) \right\}$$

($\Sigma_{x,j,l}$) depends on two terms: the motion noise (R_t), and the measurement innovation covariance matrix ($Q_{j,l}$).

The predicted measurement for landmark j ($\hat{z}_{l,j}$) is estimated from $\mu_{x_t,j,l}$, which corresponds to the estimated robot pose (using the motion model) plus a correction due to the assignment of measurement l to landmark j . This correction is proportional to two terms. The first one can be interpreted as the gain (in the same sense as the Kalman gain), and is inversely proportional to the measurement innovation covariance matrix $Q_{j,l}$, i.e. the higher the confidence in the measurement innovation (lower covariance), the higher the gain. Moreover, the gain is directly proportional to the proposal distribution covariance $\Sigma_{x,j,l}$, which means that the lower the confidence on the motion prediction (high covariance) the higher the gain (the correction due to the measurement has a high influence). On the other hand, the second term is the difference between the measurement and the prediction in the position of the landmark \bar{z}_t (Alg. 2 line 2).

C. Data association

Data association is carried out hierarchically following the Hungarian method. In the first level, measurements are assigned to the landmarks. Those measurements that have not been associated in the first level are associated to the candidate landmarks in the second stage. In this way, priority in the association is given to the landmarks, as the candidate landmarks are not reliable (many of them are created, but only a few will be initialized).

The Hungarian method is a combinatorial optimization algorithm that solves the correspondence problem in polynomial time. The method requires the construction of a cost matrix (Φ) with size $N_{z_t} \times (N_{t-1}^k + N_{z_t})$. Each element $\phi_{l,j}$ with $j \leq N_{t-1}^k$ represents the probability that measurement l is assigned to landmark j (Line 11, Alg. 2). The elements with $j > N_{t-1}^k$ represent the probability that a measurement l comes from a new landmark (P_{new}).

Given the cost matrix, the Hungarian method returns a hypothesis or ambiguity matrix where each element ($\hat{c}_{l,j}$) takes a value of 1 or 0, indicating whether or not measurement l is associated to landmark j . The ambiguity matrix fulfills the following conditions:

$$\sum_j \hat{c}_{l,j} = 1, \forall l \text{ and } \sum_l \hat{c}_{l,j} \in \{0, 1\}, \forall j \quad (2)$$

The first condition indicates that each measurement must be assigned either to a landmark or to a new landmark. The second condition reflects that a landmark can have an assigned measurement or not.

D. Robot pose update

The robot pose (Alg. 3 line 14) is sampled from a proposal distribution that considers both the motion and the observations. This proposal distribution is modelled as a Gaussian with mean μ_{x_t} and covariance Σ_{x_t} . The parameters of the Gaussian are estimated starting from the sampled pose \hat{x}_t and R_t , and iteratively adding the corrections due to the assignment of measurement z_{t,ψ_j} to landmark j (Alg. 3 lines 6 to 11).

Algorithm 3 Robot pose update algorithm

RobotPoseUpdate()

- 1: **if** $\sum_{j=1}^{N_{t-1}^k} \psi_j == 0$ **then**
- 2: $x_t^k \sim p(x_t | x_{t-1}^k, u_t)$
- 3: **else**
- 4: $\Sigma_{x,0} = R_t$
- 5: $\mu_{x,0} = \hat{x}_t$
- 6: **for** $j = 1$ to N_{t-1}^k **do**
- 7: **if** $\psi_j > 0 \ \& \ \text{type}(j) == 1$ **then**
- 8: $\Sigma_{x,j} = [H_{x,j}^T Q_{j,\psi_j}^{-1} H_{x,j} + \Sigma_{x,j-1}^{-1}]^{-1}$
- 9: $\mu_{x_t,j} = \Sigma_{x,j} H_{x,j}^T Q_{j,\psi_j}^{-1} (z_{t,\psi_j} - \bar{z}_j) + \mu_{x_t,j-1}$ +
- 10: **end if**
- 11: **end for**
- 12: $\Sigma_{x_t} = \Sigma_{x,j}$
- 13: $\mu_{x_t} = \mu_{x,j}$
- 14: $x_t^k \sim N(\mu_{x_t}, \Sigma_{x_t})$
- 15: **end if**

It is important to be careful with the order in which the landmarks are processed to generate the proposal distribution as, in each iteration, the covariance of the proposal becomes smaller and the influence of the landmark in the mean of the proposal is lower. Therefore, landmarks with lower covariance (higher confidence) are processed in first place and, in case of having the same covariance, that with a lower φ will be selected.

Finally, if none of the measurements have been assigned to previous landmarks, then the pose is generated sampling from the probability distribution of the motion model $p(x_t | x_{t-1}^k, u_t)$ (Alg. 3 line 2).

E. Landmarks update

The update of the landmarks and the estimation of \hat{w} (the contribution of each landmark to the weight of the particle) are defined in different ways depending on the type of landmark and measurement association (Alg. 4). Two different situations are possible:

- Landmarks with assigned measurements (Alg. 4 lines 3-18).
- Landmarks not seen in the current iteration (Alg. 4 lines 18- 25). There are two possibilities:
 - The landmark is outside the perceptual range of the sensor.
 - The landmark is inside the perceptual range of the sensor.

For the first situation, $i_{j,t}^k$ is incremented as the landmark has been detected. The update of the landmarks follows the standard EKF update process: first the Kalman gain K is obtained (Alg. 4 lines 8) using the previous landmark covariance $\Sigma_{j,t-1}^k$ and the measurement innovation covariance matrix \tilde{Q}_j . Although the measurement innovation covariance matrix \tilde{Q}_j was estimated previously, it must be estimated again because the Jacobian of the measurement model with respect to the measurement model variables $\tilde{H}_{m,j}$ depends of the robot pose and, now, the estimated pose (x_t^k) is more reliable than the predicted pose \hat{x}_t . The same applies to the prediction of the measurement \tilde{z}_j (Alg. 4 line 5).

Then, the mean $\mu_{j,t}^k$ is updated proportionally to the gain and the difference between the measurement and its prediction. A high Kalman gain means that the confidence in the update is high. This can occur if the previous landmark covariance was high and also, if the measurement innovation covariance is low (high confidence).

The contribution of the landmark to the weight of the particle (\hat{w}) represents the probability of association between the measurement z_{t,ψ_j} and the landmark j . It is calculated from a Gaussian distribution with mean \tilde{z}_j (predicted measurement) and covariance L . This covariance is proportional to the noise motion (R_t) and \tilde{Q}_j .

When the landmark has not been associated to any measurement, its mean and covariance remains unchanged. Moreover, if the landmark is in the perceptual range, the counter $i_{j,t}^k$ is decremented (Alg. 4 line 22). The weight \hat{w} is estimated based of the probability of visibility of the landmark (p_{in}). Finally, the weight of each particle w^k is calculated as the product over all the weights for each landmark in the map (Alg. 4 line 26), as we assume independence among the landmarks.

F. Landmarks types and initialization

The generation of new landmarks from candidate landmarks, and the modification of the type of each landmark takes place at the end of the algorithm, in functions `updateCandidateLandmarks()` and `updateLandmarksType()`:

1) `updateCandidateLandmarks()`: The candidate landmarks are updated with the measurements that were not associated to the landmarks of the map. The correspondences of each candidate landmark with the features is decided on the second

Algorithm 4 Landmarks update algorithm

LandmarksUpdate()

```

1:  $w^k = 1$ 
2: for  $j = 1$  to  $N_{t-1}^k$  do
3:   if  $\psi_j > 0$  then
4:      $i_{j,t}^k = i_{j,t-1}^k + 1$ 
5:      $\tilde{z}_j = h(\mu_{j,t-1}^k, x_t^k)$ 
6:      $\tilde{H}_{m,j} = \nabla_{m_j} h(\mu_{j,t-1}^k, x_t^k)$ 
7:      $\tilde{Q}_j = Q_{\psi_j,t} + \tilde{H}_{m,j} \Sigma_{j,t-1}^k \tilde{H}_{m,j}^T$ 
8:      $K = \Sigma_{j,t-1}^k \tilde{H}_{m,j}^T \tilde{Q}_j^{-1}$ 
9:      $\mu_{j,t}^k = \mu_{j,t-1}^k + K(z_{t,\psi_j} - \tilde{z}_j)$ 
10:     $\Sigma_{j,t}^k = (I - K \tilde{H}_{m,j}) \Sigma_{j,t-1}^k$ 
11:    if  $type(j) == 1$  then
12:       $\tilde{H}_{x,j} = \nabla_{x_t} h(\mu_{j,t-1}^k, x_t^k)$ 
13:       $L = \tilde{H}_{x,j} R_t \tilde{H}_{x,j}^T + \tilde{Q}_j$ 
14:
15:       $\hat{w} = (2\pi)^{-\frac{Dim(L)}{2}} |L|^{-\frac{1}{2}}$ 
16:       $\exp\left\{-\frac{1}{2}(z_{t,\psi_j} - \tilde{z}_j)^T L^{-1} (z_{t,\psi_j} - \tilde{z}_j)\right\}$ 
17:    else
18:       $\hat{w} = 1$ 
19:    end if
20:  else
21:     $\mu_{j,t}^k = \mu_{j,t-1}^k$ 
22:     $\Sigma_{j,t}^k = \Sigma_{j,t-1}^k$ 
23:     $\hat{w} = 1 - p_{in}$ 
24:    if  $\mu_{j,t-1}^k$  is inside perceptual range of  $x_t^k$  then
25:       $i_{j,t}^k = i_{j,t-1}^k - 1$ 
26:    end if
27:  end if
28:   $w^k = w^k \cdot \hat{w}$ 
29: end for
30:  $\{C_{1,t}^k, \dots, C_{\eta_t^k,t}^k\} =$ 
31:   updateCandidateLandmarks ( $z_t, \Psi_t^k, \{C_{1,t-1}^k, \dots, C_{\eta_{t-1}^k,t-1}^k\}$ )
32:  $\{B_{1,t}^k, \dots, B_{N_t^k,t}^k\}, \{C_{1,t}^k, \dots, C_{\eta_t^k,t}^k\} =$ 
33:   updateLandmarksType ( $\{B_{1,t-1}^k, \dots, B_{N_{t-1}^k,t-1}^k\}, \{C_{1,t}^k, \dots, C_{\eta_t^k,t}^k\}$ )

```

level of the data association process (see Sec. III-C), based on the probability $p(z_{t,l}|x_t, C_{j,t-1})$. This probability is obtained from a Gaussian distribution with mean $\hat{z}_{l,t}$ and covariance $Q_{j,l}$.

Given the correspondences obtained in the data association, the following situations may occur:

- Candidate landmark (C_j^k) has an assigned measurement. The measurement is added to the candidate landmark, i_j^k is increased, and the algorithm checks if initialization of C_j^k as a landmark is possible.
- The measurement belongs to a new candidate landmark. The measurement is added to C_j^k and i_j^k is set to 1.
- Candidate landmark (C_j^k) has no assigned measurements. Update the value of $i_j^k = i_{j,t-1}^k - I_{not}$, where I_{not} is the

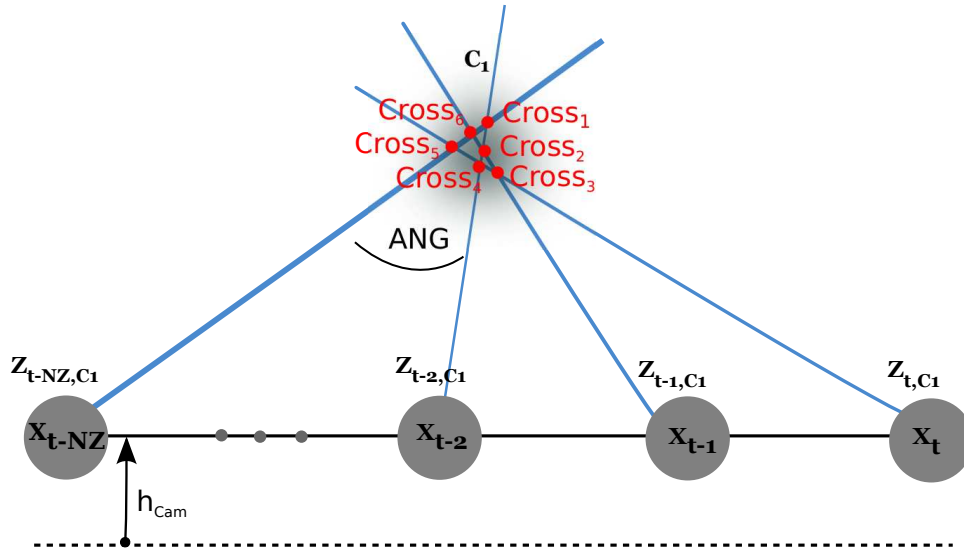


Fig. 3. Landmarks initialization.

number of consecutive iterations in which the candidate landmark was not observed. If i_j^k is not over 0, then delete the candidate landmark.

Our sensor is bearing-only and, thus, from a single measurement only angles φ and θ can be obtained for each feature. However, in order to estimate the 3D position of a landmark the distance to it (ρ) is also necessary. Therefore, at least NZ_{Min} measurements of the landmark must be taken from poses of the robot that are far enough from each other (Fig. 3). A candidate landmark (C_j^k) will become a landmark if it fulfills the following requirements:

- 1) $NZ_j^k \geq NZ_{Min}$, where NZ_j^k is the number of measurements associated along time to candidate landmark C_j^k , and NZ_{Min} is a threshold.
- 2) Of all the calculated cross-points at least $NCROSS_{VALID}$ of them are valid. A cross-point is valid if it fulfills the following properties:
 - a) The measurements used to obtain the cross-point were taken from robot poses that are separated by an angle $ANG > ANG_{Min}$.
 - b) The height of the cross-point is over the height of the camera.
 - c) The probability of all the measurements (associated to the candidate landmark) and the cross-point is over P_{New} .
- 3) One of the valid cross-points has been generated from the current measurement z_t .

If the candidate landmark is initialized, its mean is set to the pose of the valid cross-point with the highest probability over all measurements, and its covariance is set to the default initial covariance Σ_0 .

2) *updateLandmarksType()*: To add more reliability to the system, landmarks in the map are classified into types I and II. A landmark will be type I if has been initialized with measurements taken from robot poses whose distances in the XY plane from the valid cross-point are lower than D_{Min} . Those landmarks that do not meet this condition will be type

II, although they have the same priority as type I landmarks for data association.

The reliability of type II landmarks is lower than type I landmarks, as they were initialized from measurements in which the pose of the robot and the pose of the landmark were far away, and this can cause erroneous initial positions. Thus, they are not taken into account to estimate the pose of the robot (Alg. 3, lines 6 to 11) or the weights of the particles (Alg. 4, lines 11 to 17).

At each iteration, *updateLandmarksType()* checks if the landmarks of type II fulfill the requirements of type I landmarks. Moreover, all landmarks whose value of i_j^k is negative will be deleted from the map.

IV. EXPERIMENTAL RESULTS



Fig. 4. Test environment.

The proposed SLAM algorithm has been validated in a sports hall (Fig. IV) with a *Pioneer 3-DX* robot, and the landmarks (lights on the ceiling) were placed at a height of 6.5 m. The test environment has big windows that modify

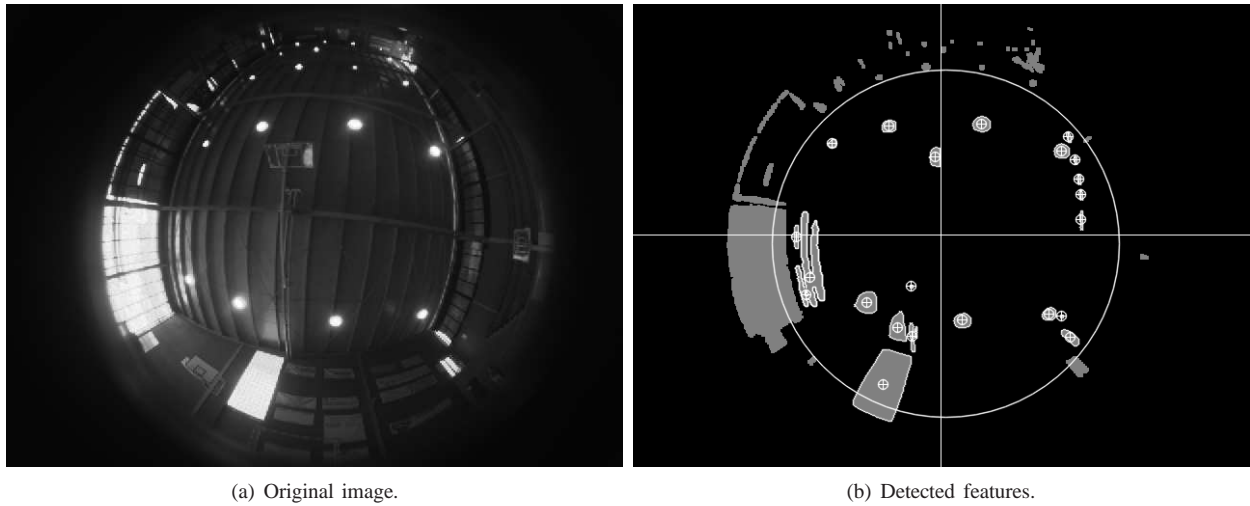


Fig. 5. Features detection.

the lightning conditions, occluding the landmarks that were situated near them. The onboard omnidirectional camera was placed at 1.8 m over the floor, in order to minimize the occlusions due to people and a passband infrared filter was used to make features extraction easier (Fig. IV). The configuration parameters of the algorithm took the following values in all the experiments: $NZ_{Min} = 3$, $NCROSS_{VALID} = 5$, $ANG_{Min} = 1.22\text{ rad}$ (7°), $D_{Min} = 8.0\text{ m}$, $\Sigma_0 = [0.0025\ 0.0; 0.0\ 0.0025]$ and $P_{New} = \frac{1}{2\pi} |Q_{l,t}|^{-\frac{1}{2}} \exp\{-\frac{64}{2}\}$. The number of particles was set to 10, as this size is sufficient to warrant the stability of the system and to achieve accurate results.

Fig. 6 shows the result of one of the tests¹. The distance travelled by the robot was over 85 m and the area of the environment was $24 \times 24\text{ m}^2$. The camera captured 320 images at 1 Hz , and the linear and angular velocities of the robot were limited to 0.30 m/s and 0.52 rad/s respectively. Fig. 6(a) shows the estimated trajectory and the obtained map and, also, the real trajectory (*GT robot*) and the real map (*GT map*). At the beginning of the trajectory (steps 0-80) the error (Fig. 6(b)) increases as the landmarks have a high covariance due to a recent initialization. Once the landmark positions are stabilized (step 100), the error in the position of the robot is reduced. From steps 100 to 180 the error remains stable, as the system detects landmarks in all the directions. Then, the error starts to grow because the robot loses the references to its right side, as landmarks 13 and 14 are not initialized yet. The consequence of the increase in the position error is that landmarks 13 and 14 are poorly initialized. The robot is able to correct its pose when it sees again previous and well placed landmarks (steps 230-270), and the error goes down near zero. Next, as the covariances of landmarks 13 and 14 reduce, they get influence in the correction of the pose of the robot, and the error increases again as their positions at that moment are still incorrect (steps 270-300). On the other hand, the error in the angle of the robot is under 0.087 rad (5°) in most of the time steps of the experiment.

ID	SLAM map (m)			GT map (m)			Error (m)			
	x	y	z	x	y	z	x	y	z	xyz
5	13.53	-5.61	6.80	13.5	-5.5	6.5	0.03	-0.113	0.299	0.32
6	13.61	-0.73	6.80	13.5	-1	6.5	0.11	0.2723	0.3	0.42
7	13.59	4.01	6.83	13.5	3.5	6.5	0.09	0.508	0.325	0.61
8	13.60	8.92	6.98	13.5	8.0	6.5	0.1	0.924	0.481	1.05
9	22.56	-5.43	6.54	22.5	-5.5	6.5	0.06	0.07	0.039	0.10
10	22.51	-0.98	6.48	22.5	-1	6.5	0.01	0.0232	-0.016	0.03
11	22.63	3.79	6.69	22.5	3.5	6.5	0.13	0.291	0.192	0.37
12	22.73	8.58	6.85	22.5	8.0	6.5	0.23	0.579	0.353	0.72
13	30.87	-5.92	6.39	31.5	-5.5	6.5	-0.63	-0.424	-0.114	0.77
14	31.18	-1.28	6.48	31.5	-1	6.5	-0.32	-0.278	-0.025	0.42

TABLE I
MAP ERROR

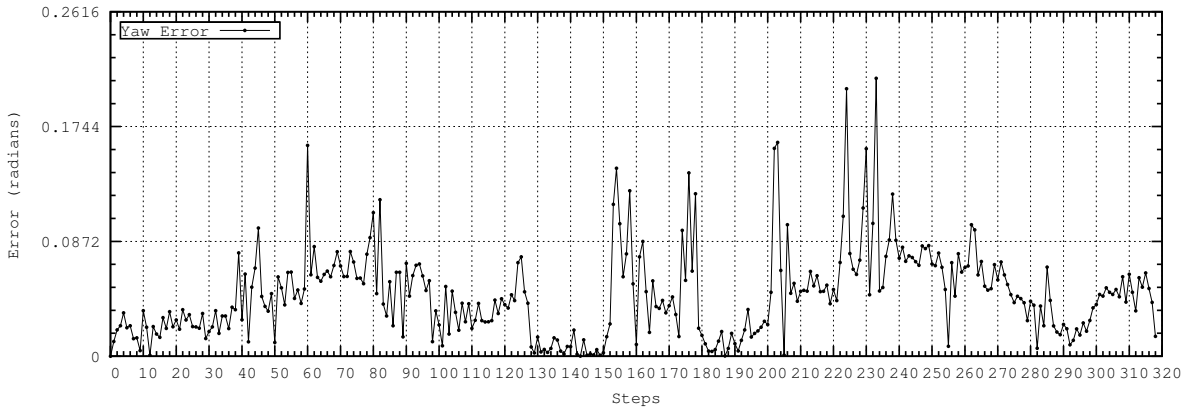
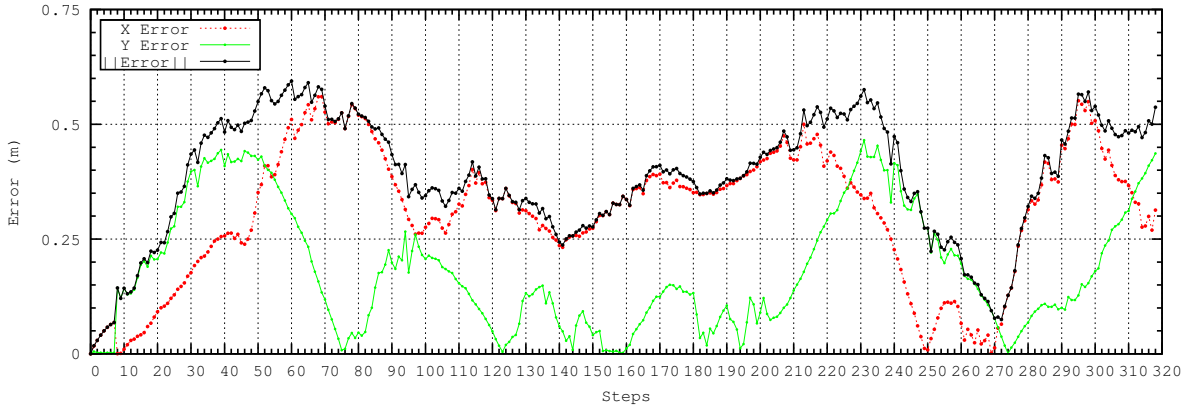
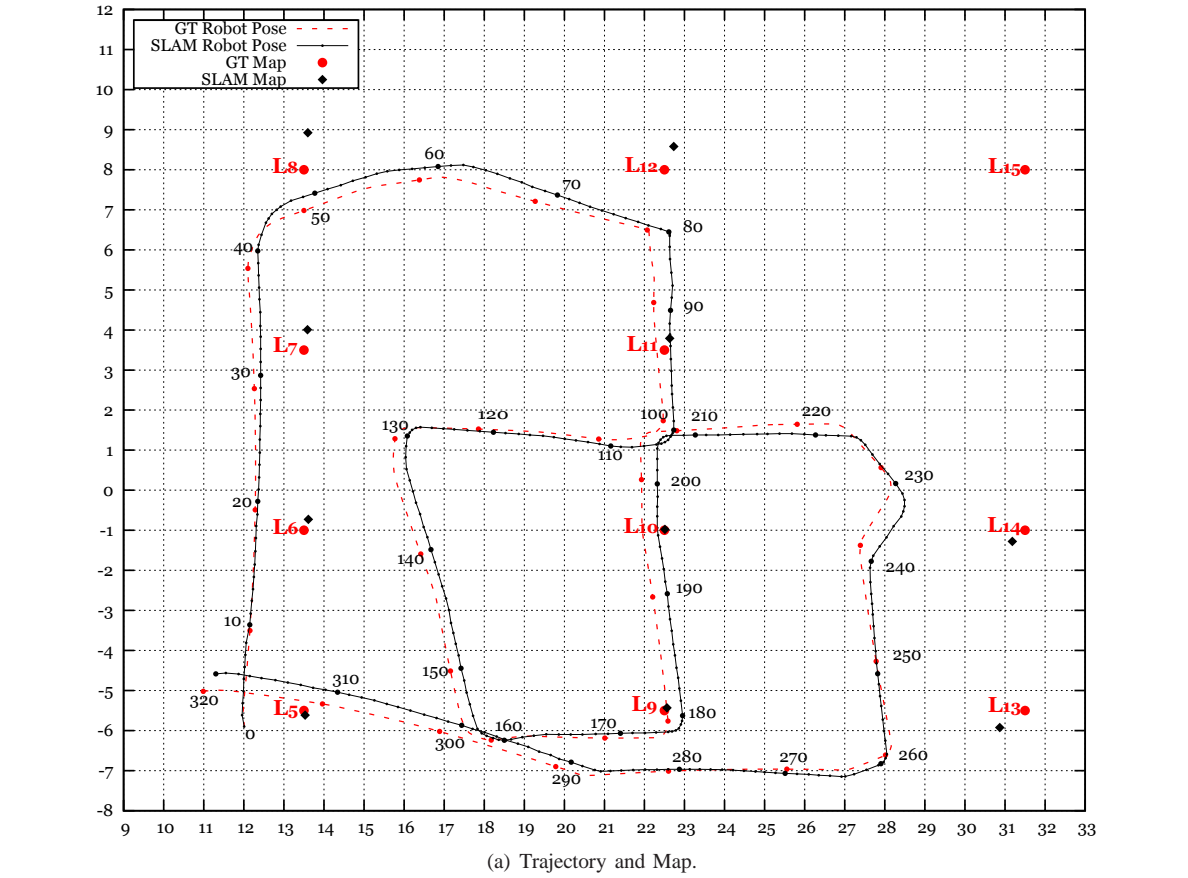
In this test, the robot returned 3 times to the same position (steps 205, 298, 318) and, in all the cases, the robot recognized the landmarks and closed the loop. The final map and its comparison with the real map is detailed in Table I. The mean and maximum errors in the position of the landmarks are 0.48 m and 1.05 m . These values reflect the performance of the SLAM algorithm as, although the sensor is bearing-only, and the landmarks are detected far away from the camera (always with at a distance over 4.5 m), their positions are quite accurate.

This experiment was run 10 times, each one with a different seed. Fig. 7 shows the average and maximum errors for each seed, together with the mean and standard deviation of all the executions. These values reflect that the proposed SLAM algorithm is able to reliably estimate the pose of the robot and the map, independently of the randomness due to the sampling steps (Alg. 3 line 14, and Alg. 1 line 9).

A. Comparison between Hungarian and Maximum Likelihood data associations

The same experiment was executed replacing the Hungarian algorithm with ML data association [26]. The results are shown in Figure 8. The errors in angle are slightly worse in ML, but the errors in position are much higher for ML in comparison with the Hungarian association: the average error

¹The associated video can be downloaded from <http://www.gsi.dec.usc.es/mucientes/videos/JOPHA13-SLAM.mp4>



(b) Errors in position and angle.

Fig. 6. Trajectory, map and errors of one of the experiments.

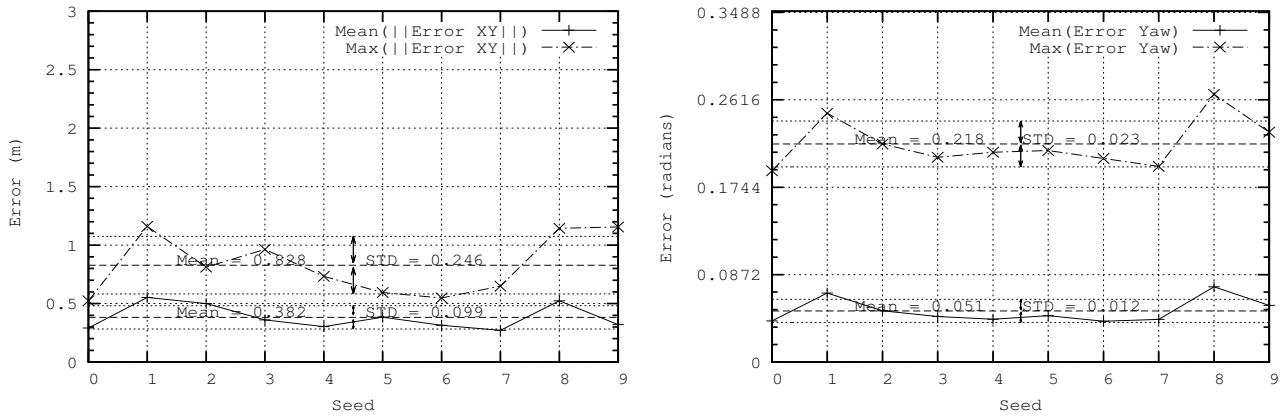


Fig. 7. Mean and maximum errors for 10 runs with Hungarian data association.

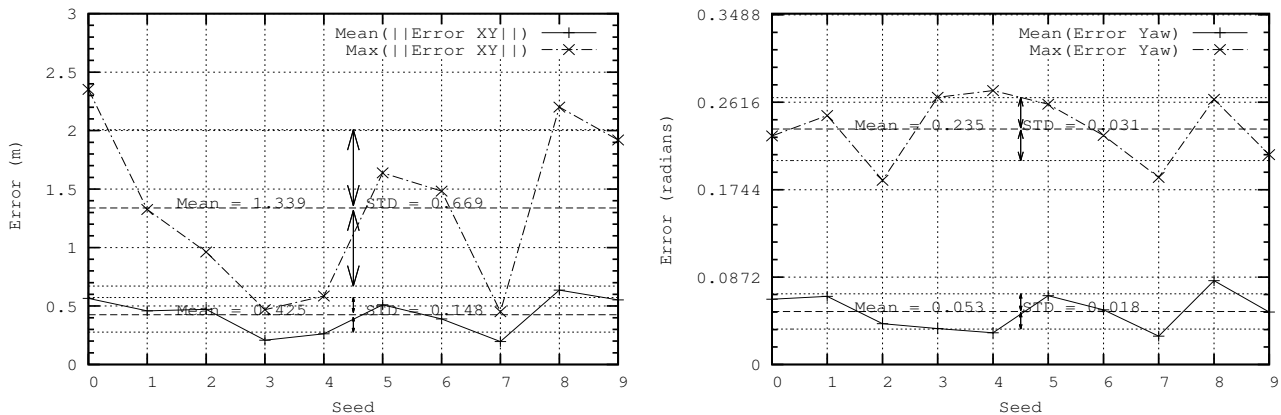


Fig. 8. Mean and maximum errors for 10 runs with ML data association.

in position is a 13% higher, and the maximum error in position is a 61% worse. Also, the standard deviation for position error is much higher for ML association. Moreover, the maximum errors with ML association are at the end of the trajectory, as the algorithm is unable to close the last loop. As a result, new landmarks are created in the bottom left part of the map Fig. 9, and the robot cannot recover its true location. Hungarian association consistently closes all the loops for all the seeds, while ML association fails to close the loops the 60% of the executions.

V. CONCLUSIONS

A SLAM algorithm, based on FastSLAM, using omnivision has been presented. Our system uses a bearing-only sensor, and the landmarks are indistinguishable. The main novelties of our proposal are the hierarchical data association based on the Hungarian algorithm and the way the landmarks are initialized.

Experiments have shown a great accuracy, both in the pose of the robot and in the map, although the limitations of the bearing-only sensor and the distance between the robot and the landmarks. Also, we have evaluated the robustness of the algorithm through several runs with different seeds, obtaining in all the experiments good results. Moreover, we have studied the influence of the use of a global data association (Hungarian

algorithm) in comparison with a local data association (Maximum Likelihood), showing that bearing-only visual SLAM with this type of landmarks requires a global data association to solve the SLAM problem reliably and accurately.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Economy and Competitiveness under grants TIN2011-22935 and TIN2009-07737 and by the Galician Government (Consolidation of Competitive Research Groups, Xunta de Galicia ref. 2010/6). Manuel Mucientes is supported by the *Ramón y Cajal* program of the Spanish Ministry of Economy and Competitiveness.

REFERENCES

- [1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002.
- [2] S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Autonomous Robots*, vol. 5, pp. 253–271, 1998.
- [3] J. A. Castellanos and J. D. Tardós, "Mobile Robot Localization and Map Building: A Multisensor Fusion Approach," in *Proceedings of the International Symposium on Experimental Robotics*, 1999, pp. 173–178.

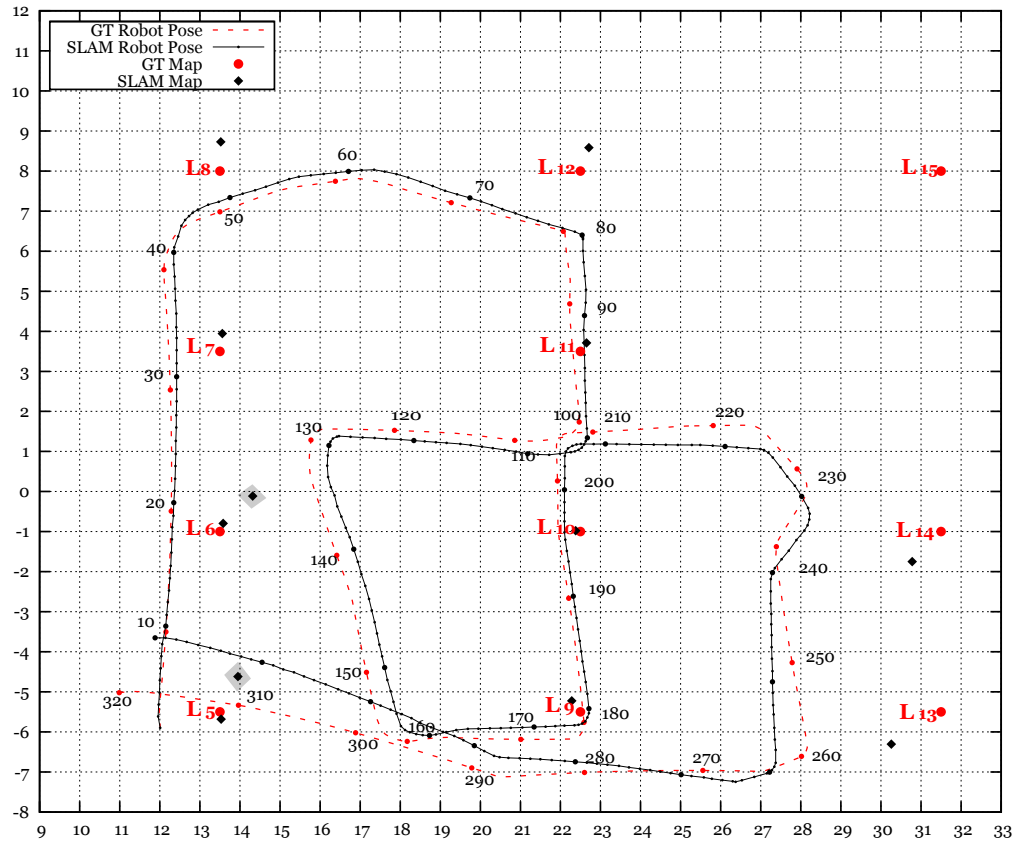


Fig. 9. Trajectory and map for ML data association. The duplicated landmarks are highlighted in gray

- [4] J. J. Leonard and H. F. Durrant-White, "Simultaneous map building and localization for an autonomous mobile robot," in *Proceedings of the IEEE/RSJ International on Intelligent Robots and Systems (IROS), Workshop on Intelligence for Mechanical Systems*, Osaka, Japan, 1991, pp. 1442–1447.
- [5] J. D. Tardos, J. Neira, P. M. Newman, and J. J. Leonard, "Robust Mapping and Localization in Indoor Environments Using Sonar Data," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.
- [6] A. J. Davison and D. W. Murray, "Mobile Robot Localisation using Active Visual Sensing," in *Proceedings of the Fifth European Conference on Computer Vision*, 1998, pp. 809–825.
- [7] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.
- [8] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM Algorithm for Robust Localization and Mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 24–29.
- [9] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 2006, pp. 534–560, 2010.
- [10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2003.
- [11] H. Andreasson, T. Duckett, and A. Lilienthal, "Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4096–4101.
- [12] E. Eade and T. Drummond, "Scalable Monocular SLAM," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 469–476.
- [13] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, "Real time data association for fastslam," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, sept. 2003, pp. 412–418.
- [14] M. Montemerlo and S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Springer, 2007.
- [15] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [16] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [17] J. S. I. Ortega, T. Lemaire, M. Devy, S. Lacroix, and A. Monin, "Delayed vs undelayed landmark initialization for bearing-only SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on SLAM*, 2005.
- [18] N. Kwok and G. Dissanayake, "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 736–741.
- [19] T. Lemaire and S. Lacroix, "SLAM with Panoramic Vision," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 91–111, 2007.
- [20] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1403–1410.
- [21] M. Deans and M. Hebert, "Invariant filtering for simultaneous localization and mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 1042–1047.
- [22] C. Schlegel and S. Hochdorfer, "Bearing-Only SLAM with an Omnicam: An Experimental Evaluation for Service Robotics Applications," *Autonome Mobile Systeme*, pp. 99–106, 2005.
- [23] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth to Depth Conversion for Monocular SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2778–2783.
- [24] C. Gamallo, C. V. Regueiro, P. Quintia, and M. Mucientes, "Omnivision-based KLD-Monte Carlo Localization," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 295–305, 2010.
- [25] H. Bakstein and T. Pajdla, "Panoramic Mosaicing with a 180 field of view lens," in *Proceedings of the Third workshop on Omnidirectional Vision*, 2002, pp. 60–67.
- [26] C. Gamallo, C. V. Regueiro, and M. Mucientes, "A FastSLAM Algorithm for Omnivision," in *Proceedings of the XIII Workshop of Physical Agents*, 2012, pp. 7–14.