A FastSLAM Algorithm for Omnivision

Cristina Gamallo, Manuel Mucientes and Carlos V. Regueiro

Abstract—**Omnidirectional cameras have a wide field of view, which makes them specially suitable for Simultaneous Localization and Mapping (SLAM) tasks. In this paper, we present a proposal for SLAM based on the well-known FastSLAM algorithm [1]. Our approach uses omnivision to detect the lights placed on the ceiling of indoor environments. As the sensor is bearing-only and the landmarks are indistinguishable among each other, the proposal includes a hierarchical data association method based on maximum likelihood and a delayed initialization of the landmarks. The proposal has been tested on a real environment with a *Pioneer 3-DX* robot**

Index Terms—**Simultaneous Localization and Mapping; FastSLAM; Omnidirectional camera; Omnivision;**

I. Introduction

TWo fundamental aspects in mobile robotics are the location of the robot and the construction of the map of the environment. The two tasks are mutually dependent on each other, i.e., mapping an environment requires a correct localization of the robot but, also, a precise positioning requires the existence of a map. In the field of robotics, the resolution of the two tasks simultaneously is known as the Simultaneous Localization and Mapping (SLAM) problem.

In the last decade the problem of SLAM has attracted the attention of many researchers in the field. The first proposals were based on Extended Kalman Filters (EKFs) using range sensors like laser range scanners [2], [3] or ultrasound sensors [4], [5], and they were prepared to run in static environments. The use of cameras to solve the SLAM problem, known as visual SLAM, is a more recent line of research. The first work 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. In the last years, an increasing attention has been paid to visual SLAM systems.

One of the main areas of research in visual SLAM has been stereo vision [7], [8], [9]. The stereo cameras can provide 3D information into a single measurement, so the traditional SLAM algorithms can 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 lets the detection of very far objects. However, they are bearing-only sensors, that is, they do not provide information about distance or depth. Therefore, a mechanism to estimate the 3D position of the landmarks needs to be introduced.

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 maximum likekihood, 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, a short introduction to bearing-only SLAM is presented. Then, Sec. III describes the proposed algorithm and Sec. IV presents the experimental results. Finally, Sec. V points out the conclusions and future work.

II. Bearing-Only SLAM

Monocular cameras are sensors that only provide information on the orientation of the objects detected in the image. As depth information is not avaliable, 3D positions of objects cannot be known with a single capture (image). This problem is known as landmark initialization in bearing-only SLAM, and the approaches to the problem can be grouped in delayed an undelayed solutions [10]. For example, in [11] an undelayed solution was proposed. The initial state of the landmarks was approximated with a sum of Gaussians that augment the state of the EKF. Most of the approaches present a delayed initialization mechanism, which requires the use of several images to estimate the 3D positionning of the landmark. [12] present a landmark initialization algorithm based also in a sum of Gaussians but without including this information in the state vector. Finally, in [13] a particle filter to estimate the initial position of the landmarks was used.

Omnivision cameras have also been used on visual SLAM. This kind of cameras have a very wide FOV, so they can track all the detected features over long distances. Therefore, the initialization estimation process is very well conditioned by the numerous observations of the same landmarks. [14] is the first work, as far as we know, that used this type of camera. The proposal is based on the relations between the beacons to estimate the pose of the robot and the odometry. 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 [15], the true location of the landmark is approximated by intersection point of two lines. [16] present a minimalist

Cristina Gamallo and Manuel Mucientes are with the Centro de Investigación en Tecnoloxías da Información (CITIUS), Universidade 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

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 $Y_{t-1}$:
         $x^k_{t-1}, \{B^k_{1,t-1}, \dots, B^k_{N^k_{t-1},t-1}\},\ \{C^k_{1,t-1}, \dots, C^k_{\eta^k_{t-1},t-1}\}$
3:     $\widehat{x}_t = g\left(x^k_{t-1},\, u_t\right)$
4:     MeasurementsLikelihood()
5:     $\Psi^k_t = dataAssociation\left(\Phi^k_t\right)$
6:     RobotPoseUpdate()
7:     LandmarksUpdate()
8: end for
9: $Y_t = lowVarianceSampling\left(\widehat{Y}_t, w^k\right)$
```

approach based on a topological map for environments of medium to large size.

Althoug most of the visual SLAM approaches are based on EKFs [13], [15], [17], there are also approaches based on FastSLAM [18], on decoupling the pose error from the map error [14], 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 [19], 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 maximum likelihood**. 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 a maximum likelihood approach.
- **Initialization of the landmarks**. We propose a initialization mechanism for the landmarks in which the 3D position of each landmark will be 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 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^k_{t-1}), a map of N^k_{t-1} landmarks $\{B^k_{1,t-1},\, \dots,\, B^k_{N^k_{t-1},t-1}\}$, and the set of η^k_{t-1} candidate landmarks $\{C^k_{1,t-1},\, \dots,\, C^k_{\eta^k_{t-1},t-1}\}$. A landmark j is defined as Gaussian with mean $\mu^k_{j,t-1}$, covariance $\Sigma^k_{j,t-1}$ and number of times detected ($i^k_{j,t-1}$). Each candidate landmark is a set of measurements $\{z^k_{t-n},\, \dots,\, z^k_{t-1}\}$ 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 to 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 lines 5).
  - **Robot pose update.** The mean and covariance of the proposal distribution will be calculated through 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 with 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 ($\widehat{Y}_t$) 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 that has been used 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 [20]. 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 ($z_{l,t}$), where each measurement is given by $(\,\varphi_{l,t},\, \theta_{l,t})$

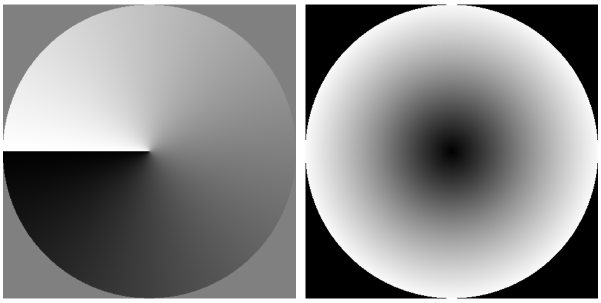
The used camera follows a projection model developed by Pajdla and Bakstein [21] that indicates how a 3D point can be transformed to a pixel in a 2D image. The model is described as a function of two angles the azimuth (φ) and the elevation (θ):

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

where *a, b, c,d* are parameters of the model, (u_0, v_0) are the coordinates of the center pixel 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. 1 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.



(a) φ (white= π, black= $-\pi$) (b) θ (white= $\pi/2$, black= 0)

Fig. 1. Graphical representation of the values of φ and θ provided by the look-up table for each image pixel.

B. Measurements likelihood and data association

The set of data associations (Ψ_t^k) between the measurements and the landmarks will be establish 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 ($z_{l,j}$), the predicted measurement ($\widehat{z}_{l,j}$, Alg. 2 line 10) and the measurement innovation covariance matrix ($Q_{l,j}$) (Alg. 2 line 11).

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

Algorithm 2 Measurements likelihood algorithm

```
    MeasurementsLikelihood()
 1: for $j = 1$ to $N_{t-1}^k$ do
 2:     $\overline{z}_j = h\left(\mu_{j,t-1}^k,\, \widehat{x}_t\right)$
 3:     $H_{x,j} = \nabla_{x_t} h\left(\mu_{j,t-1}^k,\, \widehat{x}_t\right)$
 4:     $H_{m,j} = \nabla_{m_j} h\left(\mu_{j,t-1}^k,\, \widehat{x}_t\right)$
 5:     for $l = 1$ to $N_{z_t}$ do
 6:         $Q_{l,j} = Q_{l,t} + H_{m,j}\Sigma_{j,t-1}^k {H_{m,j}}^T$
 7:         $\Sigma_{x,j} = [{H_{x,j}}^T Q_{l,j}^{-1} H_{x,j} + R_t^{-1}]^{-1}$
 8:
 9:         $\mu_{x_t,l,j} = \Sigma_{x,j} {H_{x,j}}^T Q_{l,j}^{-1}\left(z_{l,t} - \overline{z}_j\right) + \widehat{x}_t$
10:         $\widehat{z}_{l,j} = h\left(\mu_{j,t-1},\, \mu_{l,j,t}^k\right)$
11:
            $\phi_{l,j} = (2\pi)^{-\frac{Dim\left(Q_{l,j}\right)}{2}} |Q_j|^{-\frac{1}{2}}$
                $\exp\left\{-\frac{1}{2}\left(z_{l,t} - \widehat{z}_{l,j}\right)^T Q_j^{-1}\left(z_{l,t} - \widehat{z}_{l,j}\right)\right\}$
12:     end for
13: end for
```

The predicted measurement for landmark j ($\widehat{z}_{l,j}$) is estimated from $\mu_{x_t,l,j}$, 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 $Q_{l,j}$, 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 $\Sigma_{x,j}$, 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 $\overline{z}_t$ (Alg. 2 line 2).

The data association is carried out hierarchically following a maximum likelihood approach. 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).

C. 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 $\widehat{x}_t$ and R_t, and iteratively adding the corrections due to the assignment of measurement $z_{\psi_j,t}$ to landmark j (Alg. 3 lines 6 to 11).

It is important to take care of the order in which the landmarks are processed to generate the proposal distribution

Algorithm 3 Robot pose update algorithm

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

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 of $Q_{l,t}$ 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 given by the motion model $p\left(x_t | x_{t-1}^k,\, u_t\right)$ (Alg. 3 line 2).

D. Landmarks update

The update of the landmarks and the estimation of $\widehat{w}$ (the contribution of each landmark to the weight of the particle) are defined in different ways depending on the type of landmark and measurement associations (Alg. 4 loop from line 2 to 27). Two different situations are possible:

- Landmarks with assigned measurements (Alg. 4 lines 3 to 18).
- Landmarks not seen in the current iteration (Alg. 4 lines 18 to 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 update process of an EKF: 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 $\widetilde{Q}_j$. Although the measurement innovation covariance matrix $\widetilde{Q}_j$ was estimated previously, we have to estimated it again because the Jacobian of the measurement model with respect to the measurement model variables $\widetilde{H}_{m,j}$ depends of the robot pose and, now, the estimated pose (x_t^k) is more reliable than the predicted pose $\widehat{x}_t$. The same applies to the prediction of the measurement $\widetilde{z}_j$ (Alg. 4 line 5).

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

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 covariance of the landmark was high, so we will pay more attention to the current measurement. Also, if the measurement innovation covariance is low (high confidence in the innovation), the gain is high.

The contribution of the landmark to the weight of the particle ($\widehat{w}$) represents the probability of the assignment of the data association ψ_j for the landmark. It is calculated from a Gaussian distribution with mean $\widetilde{z}_j$ (predicted measurement) and covariance L for the value $z_{\psi_j,t}$ (measurement assigned to the landmark). This covariance is proportional to the noise motion (R_t) and $\widetilde{Q}_j$. 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, then the counter $i^k_{j,t}$ is decremented (Alg. 4 line 22). The weight $\widehat{w}$ 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.

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

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

- Candidate landmark (C^k_j) with an assigned measurement. The measurement is added to the candidate landmark, i^k_j is increased, and the algorithm checks if initialization of C^k_j as a landmark is possible.
- The measurement belongs to a new candidate landmark. The measurement is added to C^k_j and i^k_j is set to 1.
- Candidate landmarks (C^k_j) has no assigned measurements. Update the value of $i^k_j = i^k_{j,t-1} - I_{not}$, where I_{not} is the number of consecutive iterations in which the candidate landmark was not observed. If i^k_j is not over 0, then delete the candidate landmark.

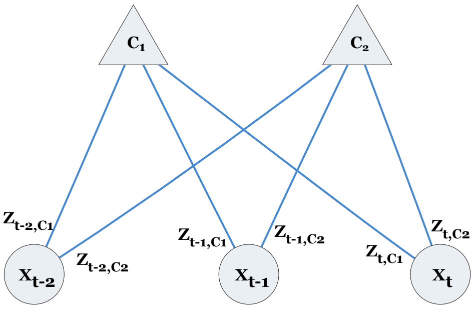


Fig. 2. Candidate landmarks.

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 taken from poses of the robot that are far enough from each other are needed (Fig. 2). A candidate landmark (C^k_j) will become a landmark if it fulfills the following requirements:

1) $NZ^k_j \geq NZ_{Min}$, where NZ^k_j is the number of measurements associated along time to candidate landmark C^k_j, 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 a minimum angle *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, but they have the same priority as type I landmarks for data association.

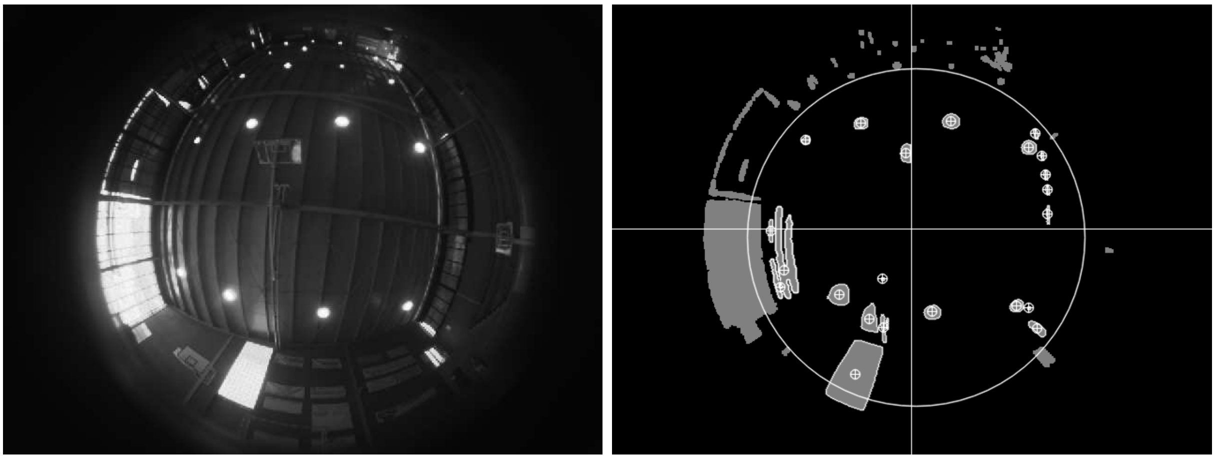
The reliability of type II landmarks is lower that type I landmarks, as the were initialized from measurements in which the pose of the robot and the pose of the landmark were far away, and this can cause erroneous initial positions. Thus, they are not taken into account to estimate the pose of the robot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

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^k_j is negative will be deleted from the map.

IV. Experimental results

The SLAM algorithm has been tested with a *Pioneer 3-DX* robot equipped with an omnidirectional camera placed at $1.8\,m$ over the floor, in order to minimize the occlusions due to people. The test environment was a sports hall (Fig. IV) with a size of $45\,x\,25\,m^2$. The landmarks used in the experiments were the lights placed at a height of $6.5\,m$. In order to make features extraction easier (Fig. IV), a passband infra-red filter was used. As can be seen, the environment also has big windows that modify the lightning conditions, occluding the artificial lights that were situated near them.

All the experiments have been executed with the following values for the parameters of the algorithm: $NZ_{Min} = 3$, $NCROSS_{VALID} = 5$, $ANG_{Min} = 7$, $D_{Min} = 8.0m.$, $\Sigma_0 = [0.0025\ 0.0; 0.0\ 0.0025]$ and $P_{New} = \frac{1}{2\pi}\left|Q_{l,t}\right|^{-\frac{1}{2}}\exp\{-\frac{64}{2}\}$. Also, images were captured at a frequency of $1.0\,Hz$, and the linear



(a) Original image.

(b) Detected features.

Fig. 4. Features detection.

Fig. 3. Test environment.

ID	SLAM map (m) x	y	z	GT map (m) x	y	z	Error (m) $\|x\|$	$\|y\|$	$\|z\|$	$\|\|xyz\|\|$
5	13.91	-5.16	5.82	13.5	-5.5	6.5	0.41	0.34	0.68	0.86
6	13.53	-0.87	6.33	13.5	-1.0	6.5	0.03	0.84	0.17	0.85
7	14.05	3.38	6.22	13.5	3.5	6.5	0.45	0.12	0.28	0.54
9	22.48	-5.98	7.12	22.5	-5.5	6.5	0.02	0.48	0.62	0.78
10	22.19	-1.02	6.69	22.5	-1.0	6.5	0.31	0.02	0.29	0.42
11	21.81	3.70	6.85	22.5	3.5	6.5	0.31	0.20	0.35	0.50
13	31.59	-5.01	6.17	31.5	-5.5	6.5	0.09	0.49	0.33	0.60
14	31.21	-0.36	6.25	31.5	-1.0	6.5	0.29	0.64	0.25	0.74

TABLE I
MAP ERROR

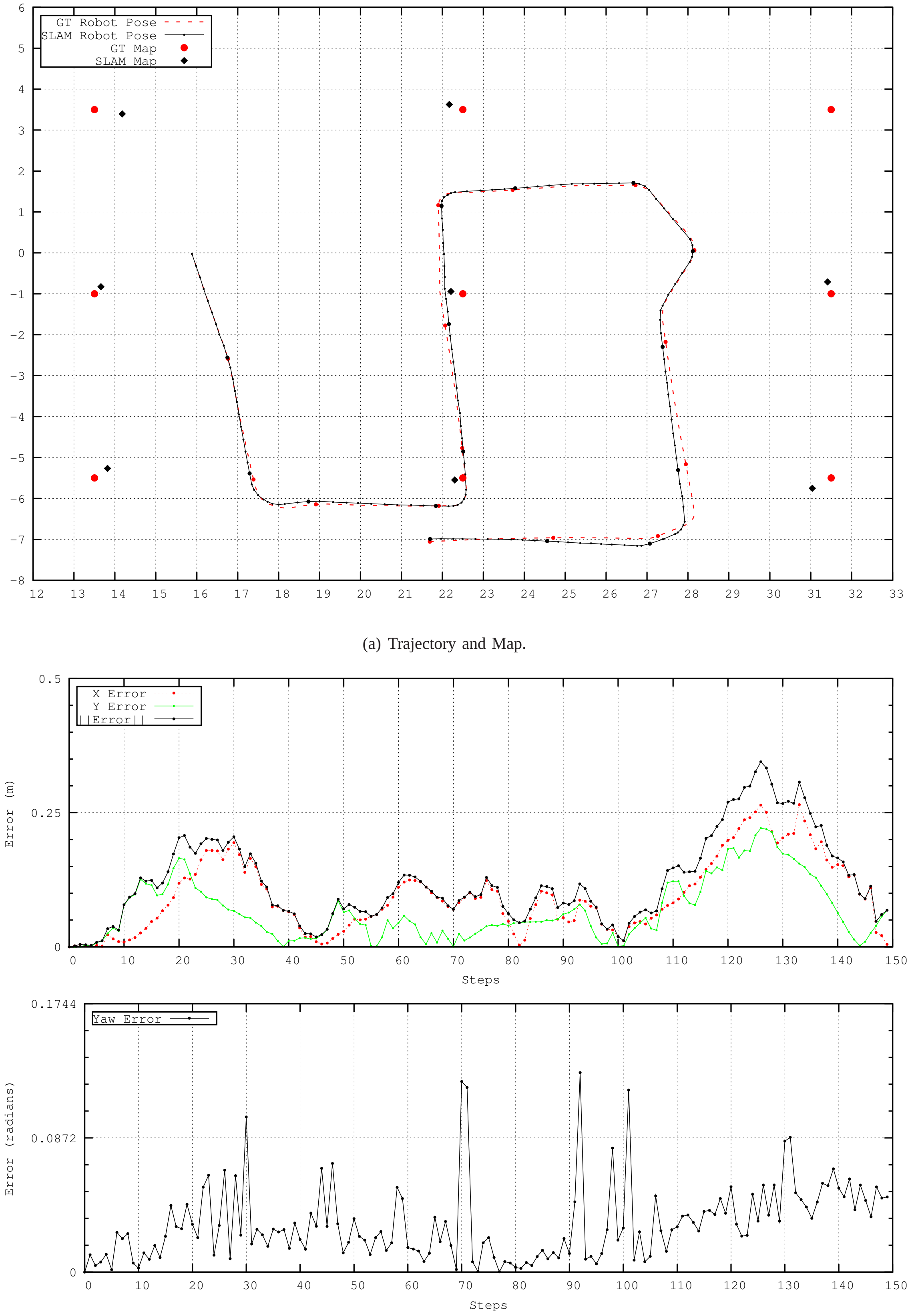
and angular velocities of the robot were limited to $0.30\,m/s$ and $0.52\,rad/s$ respectively. Finally, the number of particles was set to 10, after evaluating the performance of the algorithm with several sizes. This particles set size is sufficient to warranty the stability of the system and to achieve accurate results.

Fig. 5 shows one of the experiments[1], with a trajectory of $40\,m$ long and a total of 149 captured images. Fig. 5(a) shows the estimated trajectory and the obtained map, together with the real trajectory (GT robot) and the real map (GT map). Also, the errors in the pose of the robot (position and angle) along the experiment are also represented in Fig. 5(b). Table I shows the map obtained by the SLAM algorithm, the real map, and the errors in the position of the landmarks (for each dimension, and the total error).

As can be seen, the error in the angle of the robot is really small in most of the time steps of the experiment. Regarding the position of the robot, at the beginning the error increases as the landmarks have a high covariance due to a recent initialization. As the landmark positions are stabilized, the error in the position of the robot is reduced (central part of the experiment). When the robot arrives to the top right corner, due to the absence of a landmark, it looses the references to its left side and the error starts to grow. As a consequence, the landmarks on the right of the environment are poorly initialized, and the robot is not able to correct the position. Finally, when the robot starts to see previous well placed landmarks, the error goes down near zero. The mean error in the position of the landmarks is $0.66\,m$ and the maximum is $0.86\,m$ (table I). This 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 a distance over $4.5\,m$, as they are placed close to the ceiling), their positions are quite accurate.

In order to evaluate the robustness of the algorithm, we have executed the same experiment 10 times, each one with a different seed. This test evaluates the algorithm independently of the randomness due to the sampling of the pose of the robot (Alg. 3 line 14). The results for each seed are shown in Fig. 6. Both the average and the maximum errors in position and orientation for each run are shown. The graph also displays the mean and standard deviation for all the executions, both for the average and maximum errors. The mean of the average errors is $0.30\,m$ in position and $0.03\,rad$ in orientation, while the mean of the maximum errors is $0.67\,m$ and $0.16\,rad$ respectively. With these values, we can conclude that proposed SLAM algorithm

[1] The associated video can be downloaded from http://www.gsi.dec.usc.es/mucientes/videos/WAF12_SLAM.mp4



(a) Trajectory and Map.

(b) Errors in position and angle.

Fig. 5. Trajectory, map and errors of one of the experiments.

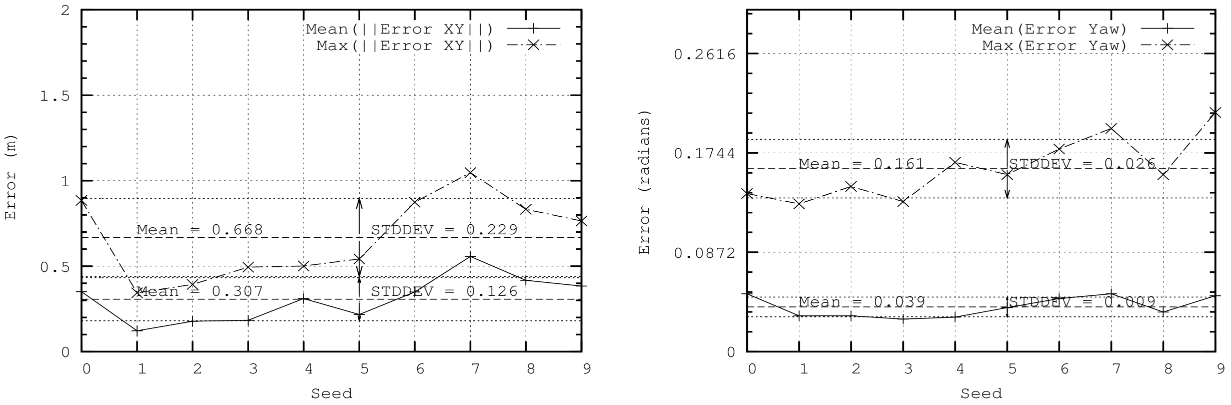


Fig. 6. Mean and maximum errors for 10 runs with different seeds.

is able to reliably estimate the pose of the robot and the map.

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 maximum likelihood 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. Moreover, we have also evaluated the robustness of the algorithm through several runs with different seeds, obtaining in all the experiments good results. As future work, we plan to improve the data association, as this stage is fundamental for a good performance of our proposal.

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness under grants TIN2011-22935 and TIN2009-07737. 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: AAAI, 2002.

[2] S. Thrun, w. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots," *Autonomous Robots*, vol. Volume 5,, pp. 253–271, July 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.

[4] J. J. Leonard and H. F. Durrant-White, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, vol. 3, 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, Apr. 2002.

[6] A. J. Davison and D. W. Murray, "Mobile Robot Localisation using Active Visual Sensing," in *Proceedings of Fifth European Conference on Computer Vision*, B. Burkhardt, Hans; Neumann, Ed., 1998, pp. 809–825.

[7] A. J. Davison and D. 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," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.*, pp. 24–29, 2005.

[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] 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 Internacional Conference on Robotics and Automation, Workshop on SLAM*, 2005.

[11] N. Kwok and G. Dissanayake, "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," in *Proceedings of The 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 2004, pp. 736–741.

[12] T. Lemaire and S. Lacroix, "SLAM with Panoramic Vision," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 91–111, Jan. 2007.

[13] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1403–1410. [Online]. Available: http://ieeexplore.ieee.org/search/selected.jsp

[14] M. Deans, "Invariant filtering for simultaneous localization and mapping," *Robotics and Automation.*, no. April, pp. 1042–1047, 2000.

[15] 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.

[16] 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 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4096–4101.

[17] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth to Depth Conversion for Monocular SLAM," in *Proceedings of The 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2778–2783.

[18] E. Eade and T. Drummond, "Scalable Monocular SLAM," in *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 469–476.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2003.

[20] C. Gamallo, C. V. C. V. Regueiro, P. Quintía, and M. Mucientes, "Omnivision-based KLD-Monte Carlo Localization," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 295–305, Mar. 2010.

[21] H. Bakstein and T. Pajdla, "Panoramic Mosaicing with a 180 field of view lens," in *Proceedings of The Third workshop on Omnidirectional Vision*, vol. 2, 2002, pp. 60–67.