

# Visual FastSLAM through Omnivision

Cristina Gamallo, Manuel Mucientes, and Carlos V. Regueiro

**Abstract**—Simultaneous Localization and Mapping (SLAM) is still an open problem in mobile robotics. In this paper a SLAM algorithm using omnivision is presented. Omnidirectional cameras have a wide field of view, thus detecting landmarks over long distances, but this also requires a good data association. Our SLAM proposal is based on the well-known FastSLAM algorithm [1]. Our main contributions are the features detection and their position estimation for an omnidirectional camera (bearing-only sensor), and also the data association process based on the Hungarian algorithm. The system has been tested on a *Pioneer 3-DX* robot equipped with an omnidirectional camera (a camera with a fish-eye lens) and a passband infrared filter. Experiments were carried out in an exposition hall of a museum, showing a good performance, despite the uneven floor which generates a swinging on the camera and increases the error in the motion commands.

## I. INTRODUCTION

In mobile robotics, the existence of a map of the environment increases the ability of the robot to implement complex tasks. Mapping an environment requires a correct localization of the robot but, also, a precise positioning requires the existence of a map. Thus, mapping and localization are coupled. The Simultaneous Localization and Mapping (SLAM) problem consists in building a map of an unknown environment from an initially unknown pose of the robot while, at the same time, the robot is localized in the environment that is being mapped.

A great effort has been dedicated to solve this problem in recent years. Many solutions have been presented, usually in 2D environments and using laser scanners. Nevertheless, several important challenges remain. One of them is operating in very crowded environments, as people moving in the surrounding of the robot produce interferences for typical sensors like laser scanners. These situations are habitual, for example, for a tour-guide robot operating in a museum, and surrounded by a group of people that are interacting with the robot. Another open issue is mapping environments with very irregular floors, where the pitch and roll angles are not negligible.

We have already proposed approaches for localization [2], [3] in such type of environments, but a map of features (landmarks) has to be provided to the system. Nevertheless, creating such a map is tedious and the resulting map is imprecise on large and/or complex environments.

Cristina Gamallo and Manuel Mucientes are with Department of Electronics and Computer Science, University of Santiago de Compostela, E-15782 Spain. [cristina.gamallo@usc.es](mailto:cristina.gamallo@usc.es), [manuel.mucientes@usc.es](mailto:manuel.mucientes@usc.es)

Carlos V. Regueiro is with Department of Electronic and Systems, University of A Coruña, E-15071 Spain. [cvazquez@udc.es](mailto:cvazquez@udc.es)

In this paper we propose a SLAM method based on the FastSLAM [4], [5], [1] algorithm but adapted to the use of an omnidirectional camera pointing to the ceiling. The camera mounts a fish-eye lens with a very wide field of view (FOV): 185 degrees. Also, the camera operates on the infrared (IR) spectrum, as it is equipped with a bandpass IR filter. The main advantage of this filter is that features are more easily detected in indoor environments because very few objects emit in the IR spectrum (for example, ceiling lights). On the other hand, the data association problem is reduced compared with the typical features used on visual SLAM.

The main contributions of the paper are: the detection of features for omnivision and the data association process. The paper is organized as follows: in the next section, a short introduction to the related work is presented. Secs. III and IV explain the FastSLAM algorithm and its adaptation to omnivision cameras. Then, some experimental results are shown in Sec. V and, finally, Sec. VI points out the conclusions and future work.

## II. RELATED WORK

In the last years, special attention has been paid to visual SLAM, as cameras provide richer information of the environment.

### A. Monocular SLAM

Monocular cameras are bearing-only sensors, that is, they do not provide information about distance or depth. Usually, *delayed* initialization of detected landmarks has to be applied in order to calculate the distance to the camera. For example, moving the robot until enough “parallax” has been achieved. The worst situation is produced by features close to the optical axis of the camera and far away from it. On the other hand, monocular cameras can detect features at very long distances which is very useful for orientation with spatial references on the horizon.

Several monocular SLAM systems have been presented for small environments [6], [7]. Most are based on Extended Kalman Filter (EKF) SLAM or FastSLAM [8], but differ in some aspects. One of them is that the points extracted from the images can be Harris corners, scale-invariant feature transform (SIFT) features or Shi-Tomasi corners. A fundamental limitation of monocular camera SLAM is that the scale of the environment can not be calculated directly. Therefore, additional information needs to be included.

Nevertheless, almost all features are locally detected and their detection strongly depends on lighting conditions, surrounding textures, and the position and orientation of the camera. These limitations have a high impact for SLAM,

mainly when closing loops, i.e., being able to detect that a place of the environment has been revisited. The narrow FOV of monocular cameras makes this identification very difficult, because the new point of view can be very different from that of previous visits.

Some recent works employ a *direct approach* to visual monocular SLAM [9] and use the image intensity as observations instead of features. The visual SLAM is formulated as a non-linear image alignment task. Their main limitations are due to changes of lighting conditions, the high computational cost, and that only planar surfaces are modeled.

### B. Stereovision SLAM

Stereo visual systems provide scale through the baseline between cameras, precisely known from a calibration process. In this way, they provide all the necessary 3D information of each detected feature.

The main drawbacks of stereovision are the limited 3D range and the mechanical fragility. While cameras can detect very far objects, stereo systems only provide good 3D information to a limited range, typically from 3m to a few tens of meters depending on the baseline. To extend range, the stereo baseline has to be increased while keeping or improving the overall sensor precision, which is contradictory.

Davison and Murray presented the first active stereo visual SLAM system [10] based on standard EKF. Another stereovision SLAM using SIFT features in a small laboratory has been presented in [11]. A dense system using Rao-Blackwellized particle filters and SIFT features was described in [12]. Saez et al. [13] present a 6-DOF stereo SLAM using a global entropy minimization algorithm. Recently, an important advance has been achieved on stereovision SLAM to cope with medium size indoor and outdoor environments [14].

Some authors have explored the integration of both monocular and stereo vision [15]. They claimed that stereovision can be seen not as a monolithic supersensor, but as two independent monocular cameras. This method can be applied to stereo systems, several different cameras, and cameras moving independently.

### C. Structure From Motion and Visual Odometry

Monocular SLAM can be compared with the structure from motion problem (SFM). In SFM, the goal is to determine, from a collection of images and up to an unrecoverable scale factor, the 3D structure of the environment and all the 6-D camera poses (position and orientation) from where the images were captured. The differences between SFM and SLAM are not only in the methods but also in the objectives. That is, similar aspects of similar problems are given different priorities.

On the other hand, in Visual Odometry (VO) the robot's ego motion has to be obtained from a sequence of images. This can be seen as a similar problem to stereovision SLAM and features have to be matched across two or more stereo pair of images. VO must work in real time because robot position is needed on-line. Advanced solutions achieve very low drift level after long distances.

### D. Omnivision SLAM

Omnivision cameras have also been used on visual SLAM. Usually, the approaches employ an additional stereo vision system for motion observation between consecutive frames. Omnivision cameras have a very wide FOV, so they can track all the detected features (close or far to the camera) over long distances. The estimation process is very well conditioned by these numerous observations of the same landmarks.

Lemaire and Lacroix [16] present an efficient and optimal *delayed* landmark initialization algorithm. Loop closures are detected through a database query that retrieves memorized images that are close to the current robot estimation.

Rybsk et al. [17] propose a method to incrementally build topological maps based on the panoramic images (“signature”) captured by the robot on each position of the traveled path. They applied the Iterated form of the Extended Kalman Filter (IEKF) and a batch-processed linearized maximum likelihood estimator.

One limitation of omnivision SLAM systems is the feature matching algorithm, as small translations and rotations cause important changes in the images. For this reason, local conditions can change dramatically, specially on features close to the optical axis. In such situations, matching algorithms produce many outliers. With classic perspective cameras, they can be discarded by checking the epipolar constraint, but with omnivision cameras this constraint is more tricky to establish.

## III. SLAM ALGORITHM

The SLAM algorithm described in this section is based on FastSLAM 2.0 [1]. This algorithm uses a Rao-Blackwellized particle filter, which represents the posterior distribution with particles for some variables and a parametric representation (such as Gaussians) for the others [18]. The main idea of the basic FastSLAM algorithm is that the pose of the robot is estimated with a particle filter, while each feature in the map is tracked with an individual and low-dimensional EKF.

However, the main limitation of FastSLAM is that the pose of the robot is sampled taking into account only the control action. This is specially important when the accuracy of the sensors is high relative to the accuracy in control. The result is that the posterior distribution, which has been generated by resampling from the proposal distribution, is poorly matched with the proposal distribution. This drawback is solved by FastSLAM 2.0, as poses are sampled based on motion and the observations. Therefore, the matching between the proposal and posterior distributions is improved.

Our SLAM proposal is shown in Algs. 1 and 2. The main contributions of our approach are:

- The detection of the features and the estimation of their position ( $z_{l,t}$ ) for an omnidirectional camera (a camera with a fish-eye lens). The process will require to approximate the model of the camera to relate a pixel with the 3D coordinates. Moreover, as this camera is a bearing-only sensor, the 3D position of each feature will be obtained through several consecutive detections from different positions.

- The data association process, based on the Hungarian algorithm, which looks for the best association among landmarks and measurements. This differs from the maximum likelihood approach that tries to maximize the individual associations (selects the best measurement for each landmark) instead of the global association.
- Also, due to the extension of the algorithm to multiple measurements, several small changes have been done to the basic algorithm. The most important ones are in the calculation of the pose of the robot as a weighted average of the poses obtained for each measurement, and the estimation of the weight of each particle.

The algorithm (Alg. 1) receives the set of measurements ( $z_t$ ) at the current time  $t$ , the control ( $u_t$ ) and the previous set of particles ( $Y_{t-1}$ ). The main loop of the algorithm iterates for each of the  $M$  particles in the previous set. A particle  $k$  is defined (line 3) as the pose of the robot ( $x_{t-1}^k$ ), the number of landmarks in the map ( $N_{t-1}^k$ ), and the set of landmarks defined by the mean ( $\mu_{j,t-1}^k$ ), covariance ( $\Sigma_{j,t-1}^k$ ), and number of times detected ( $i_{j,t-1}^k$ ) of each of them.

The loop from lines 5 to 20 iterates for all the landmarks in order to estimate  $\phi_{l,j}$  (line 15), i.e., the probability that measurement  $l$  corresponds to landmark  $j$ . Before this loop, the pose of the robot needs to be estimated using the motion model (line 4). With this estimation, a prediction of the measurement for landmark  $j$  is calculated ( $\bar{z}_j$ ). 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, the landmark innovation covariance matrix ( $Q_j$ ) is calculated.

As has already been mentioned, the improvement of FastSLAM 2.0 over the basic FastSLAM algorithm is that the pose of the robot is sampled from a proposal distribution that considers both the motion and the observations. This proposal distribution is modeled as a Gaussian with mean  $\mu_{x_t,l,j}$  and covariance  $\Sigma_{x,j}$ . The covariance of the proposal distribution depends on two terms: the motion noise ( $R_t$ ), and a covariance that is inversely proportional to the landmark innovation covariance matrix.

The estimation of the mean of the proposal distribution and, therefore, the probabilities ( $\phi_{l,j}$ ) depends on the measurements (lines 11 to 19). The mean ( $\mu_{x_t,l,j}$ ) corresponds to the estimated pose of the robot (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 second one is the difference between the measurement and the prediction in the position of the landmark. On the other hand, the first one can be interpreted as the gain (in the same sense as the Kalman gain), and is inversely proportional to the landmark innovation covariance matrix ( $Q_j$ ), i.e. the higher the confidence in the landmark 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 (high covariance) the higher the gain (the correction due to the measurement has a high influence).

It is worthy to mention that the proposal distribution has been obtained with the motion model and the measurement model, but only using the association of a landmark with a measurement, as the best data association is already unknown (it requires the calculation of all the probabilities,  $\phi_{l,j}$ ). In an ideal situation with known correspondences between landmarks and measurements, the proposal distribution should include all the landmarks and measurements, and not just one. As a result of the unknown correspondences,  $j \times l$  sampled poses of the robot ( $x_{l,j,t}^k$ ) are generated.

The calculation of the probability that measurement  $l$  corresponds to landmark  $j$  ( $\phi_{l,j}$ ) depends on the predicted measurement. This was already estimated (line 6), but using the pose of the robot predicted by the motion model. However, a better estimation of the predicted measurement is given if the sampled pose of the robot is used (line 14). Finally, using  $\hat{z}_{l,j}$ ,  $\phi_{l,j}$  is calculated using as covariance matrix the measurement innovation.

#### A. Data Association

The data association process requires the construction of a cost matrix ( $\Phi$ ). This is an  $N_{z_t} \times (N_{t-1}^k + N_{z_t})$  matrix in which each element ( $\phi_{l,j}$ ) with  $j \leq N_{t-1}^k$  represents the probability that measurement  $l$  was originated from landmark  $j$ , and the elements with  $j > N_{t-1}^k$  represent the probability that measurement  $l$  came from a new landmark. Once the cost matrix has been calculated, the best data association, i.e., that with the higher probability is determined with the Hungarian algorithm.

The Hungarian method [19] is a combinatorial optimization algorithm which solves the assignment problem in polynomial time. The method generates a hypothesis or ambiguity matrix ( $\hat{c}$ ) for which each element ( $\hat{c}_{l,j}$ ) can take a value of 1 or 0, representing the possibility that measurement  $l$  is associated to landmark  $j$  or not. The ambiguity matrix fulfills the following two conditions:

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

The first condition means that a measurement should only be assigned to a landmark, i.e., each row of the ambiguity matrix has only a one. The condition also applies to the columns, as a landmark can generate only one measurement. For these reasons,  $N_{z_t}$  columns were added to the cost matrix to represent the possibility that the  $N_{z_t}$  measurements correspond all to new landmarks. This is done in the loop at line 16, where  $p_0$  represents the probability that the measure came from a new landmark.

Once the data association has been obtained, the pose of the robot can be calculated. If none of the measurements have been assigned to the previous landmarks, then the pose is generated sampling from the probability distribution given by the motion model (line 26). On the other hand, if any of the measurements corresponds to a previous landmark, the sampled poses of the robot ( $x_{l,j,t}^k$ ) will be used to obtain the final pose (line 28). The pose is calculated as the weighted average of the sampled poses for those landmarks that existed

**Algorithm 1** SLAM algorithm based on FastSLAM 2.0 [1].

---

```

1: SLAM( $z_t, u_t, Y_{t-1}$ )
2: for  $k = 1$  to  $M$  do
3:   Get particle  $k$  from  $Y_{t-1}$ :
       $x_{t-1}^k, N_{t-1}^k, \{\langle \mu_{1,t-1}^k, \Sigma_{1,t-1}^k, i_{1,t-1}^k \rangle, \dots, \langle \mu_{N_{t-1}^k,t-1}^k, \Sigma_{N_{t-1}^k,t-1}^k, i_{N_{t-1}^k,t-1}^k \rangle\}$ 
4:    $\hat{x}_t = g(x_{t-1}^k, u_t)$ 
5:   for  $j = 1$  to  $N_{t-1}^k$  do
6:      $\bar{z}_j = h(\mu_{j,t-1}^k, \hat{x}_t)$ 
7:      $H_{x,j} = \nabla_{x_t} h(\mu_{j,t-1}^k, \hat{x}_t)$ 
8:      $H_{m,j} = \nabla_{m_j} h(\mu_{j,t-1}^k, \hat{x}_t)$ 
9:      $Q_j = Q_t + H_{m,j} \Sigma_{j,t-1}^k H_{m,j}^T$ 
10:     $\Sigma_{x,j} = [H_{x,j}^T Q_j^{-1} H_{x,j} + R_t^{-1}]^{-1}$ 
11:    for  $l = 1$  to  $N_{z_t}$  do
12:       $\mu_{x_t,l,j} = \Sigma_{x,j} H_{x,j}^T Q_j^{-1} (z_{l,t} - \bar{z}_j) + \hat{x}_t$ 
13:       $x_{l,j,t}^k \sim N(\mu_{x_t,l,j}, \Sigma_{x,j})$ 
14:       $\hat{z}_{l,j} = h(\mu_{j,t-1}^k, x_{l,j,t}^k)$ 
15:
16:       $\phi_{l,j} = |2\pi Q_j|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{l,t} - \hat{z}_{l,j})^T Q_j^{-1} (z_{l,t} - \hat{z}_{l,j})\}$ 
17:
18:      for  $e = 1$  to  $N_{z_t}$  do
19:         $\phi_{l,e+N_{t-1}^k} = p_0$ 
20:      end for
21:    end for
22:     $\hat{c} = \text{dataAssociation}(\Phi)$ 
23:     $\hat{N}_t^k = N_{t-1}^k + N_{t,new}^k$ 
24:     $N_t^k = \hat{N}_t^k$ 
25:     $w^k = 1$ 
26:    if  $\sum_{j=1}^{N_{t-1}^k} \hat{c}(j) == 0$  then
27:       $x_t^k \sim p(x_t | x_{t-1}^k, u_t)$ 
28:    else
29:       $x_t^k = \frac{\sum_{j=1}^{N_{t-1}^k} \phi_{\hat{c}(j),j} \cdot x_{\hat{c}(j),j,t}^k}{\sum_{j=1}^{N_{t-1}^k} \phi_{\hat{c}(j),j}}$ 
30:    end if

```

---

at  $t-1$  and that have been detected at  $t$ . Function  $\hat{c}(j)$  returns the measurement assigned to landmark  $j$ , with  $\hat{c}(j) = 0$  representing no assignment to a measurement.

### B. Landmarks update

The update of the mean, covariance, and number of times detected for each of the landmarks of the particle (lines 30 to 62) depends on the data association given by  $\hat{c}$ . Also, the contribution of the landmark ( $\hat{w}$ ) to the weight of the particle ( $w^k$ ) must take into account the data association. The weight of a landmark ( $\hat{w}$ ) represents the probability of the assignment of the data association  $\hat{c}$  for the landmark.

**Algorithm 2** Continued from Alg. 1: SLAM algorithm based on FastSLAM 2.0 [1].

---

```

30:   for  $j = 1$  to  $\hat{N}_t^k$  do
31:     if  $j \leq N_{t-1}^k$  then
32:       if  $\hat{c}(j) > 0$  then
33:          $K = \Sigma_{j,t-1}^k H_{m,j}^T Q_j^{-1}$ 
34:          $\mu_{j,t}^k = \mu_{j,t-1}^k + K(z_{\hat{c}(j),t} - \bar{z}_j)$ 
35:          $\Sigma_{j,t}^k = (I - K H_{m,j}) \Sigma_{j,t-1}^k$ 
36:          $i_{j,t}^k = i_{j,t-1}^k + 1$ 
37:          $L = H_{x,j} R_t H_{x,j}^T + H_{m,j} \Sigma_{j,t-1}^k H_{m,j}^T + Q_t$ 
38:
39:          $\hat{w} = |2\pi L|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{\hat{c}(j),t} - \hat{z}_{\hat{c}(j),j})^T L^{-1} (z_{\hat{c}(j),t} - \hat{z}_{\hat{c}(j),j})\}$ 
40:       else
41:          $\mu_{j,t}^k = \mu_{j,t-1}^k$ 
42:          $\Sigma_{j,t}^k = \Sigma_{j,t-1}^k$ 
43:         if  $\mu_{j,t-1}^k$  is outside perceptual range of  $x_t^k$ 
44:           then
45:              $i_{j,t}^k = i_{j,t-1}^k$ 
46:              $\hat{w} = 1$ 
47:           else
48:              $i_{j,t}^k = i_{j,t-1}^k - 1$ 
49:             if  $i_{j,t}^k < 0$  then
50:               discard feature  $j$ 
51:                $N_t^k = N_t^k - 1$ 
52:                $\hat{w} = p_n$ 
53:             end if
54:           end if
55:         end if
56:       else
57:          $\mu_{j,t}^k = h^{-1}(z_{\hat{c}(j),t}, x_t^k)$ 
58:          $H_{m,j} = \nabla_{m_j} h(\mu_{j,t}^k, x_t^k)$ 
59:          $\Sigma_{j,t}^k = (H_{m,j}^T Q_t^{-1} H_{m,j})^{-1}$ 
60:          $i_{j,t}^k = 1$ 
61:          $\hat{w} = p_0$ 
62:       end if
63:        $w^k = w^k \cdot \hat{w}$ 
64:     end for
65:   end for
66:   while  $|Y_t| < M$  do
67:     draw particle  $k$  from  $\hat{Y}_t$  with probability  $\propto w^k$ 
68:     Add  $x_t^k, N_t^k, \{\langle \mu_{1,t}^k, \Sigma_{1,t}^k, i_{1,t}^k \rangle, \dots, \langle \mu_{N_t^k,t}^k, \Sigma_{N_t^k,t}^k, i_{N_t^k,t}^k \rangle\}$  to  $Y_t$ 
69:   end while

```

---

Therefore, the weight of each particle is calculated as the product over all these probabilities –weights– (line 61). Both the update of the landmarks and the estimation of  $\hat{w}$  are defined in different ways depending on the type of landmark

and measurement associations:

- 1) Measurements assigned to previous landmarks (from line 32).
- 2) Landmarks without assigned measurements (from line 39). There are two possibilities:
  - a) The landmark is outside the perceptual range of the sensor (from line 42).
  - b) The landmark is inside the perceptual range of the sensor (from line 45).
- 3) Measurements assigned to new landmarks (from line 54).

For the first situation, the update of the landmarks follows the standard update process of an EKF: first the Kalman gain is obtained using the previous landmark covariance and the measurement innovation covariance matrix ( $Q_j$ ). Then, the mean 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 have to rely on the current measurement. Also, if the measurement innovation covariance is low (high confidence in the innovation), the gain is high. Moreover,  $i_{j,t}^k$  is incremented as the landmark has been detected.

The contribution of the landmark to the weight of the particle is calculated from a Gaussian distribution with mean  $\hat{z}_{c(j),j}$  (predicted measurement) and covariance  $L$  for the value  $z_{c(j),t}$  (measurement assigned to the landmark). This covariance is proportional to the motion ( $R_t$ ) and measurement noises ( $Q_t$ ), and the previous landmark covariance.

The second and third situations (landmark without measurement) keep unchanged the mean and covariance of the landmark. If the landmark is in the perceptual range, then the counter  $i_{j,t}^k$  is decremented, and if it reaches a value under 0, the landmark is deleted. This contributes to the elimination of false landmarks and keeps their number in a reasonable range. The probability assigned to the third situation is the constant  $p_n$ . Finally, if the measurement corresponds to a new landmark, the mean is initialized from the measurement and the pose of the robot, and the covariance is proportional to the measurement noise covariance, while the contribution to the weight of the particle is also a constant ( $p_0$ ).

Finally, the new particle set ( $Y_t$ ) is generated by sampling the updated particle set ( $\hat{Y}_t$ ) with probabilities proportional to the particle weights ( $w^k$ ).

#### IV. MEASUREMENT MODEL

The sensor model that has been used is based on feature extraction from the images obtained by an omnidirectional camera (a camera with a fish-eye lens). The landmarks are the lights placed on the ceiling of the environment.

##### A. Features extraction

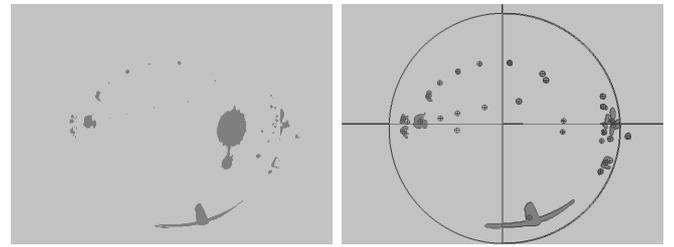
Due to the special characteristics of the landmarks, the features extraction process can be improved incorporating a passband infrared filter to the camera. The process for features detection consists of five steps: acquisition, preprocessing, segmentation, recognition and features extraction

[20], [21]. The output of the process is an array of features. In the preprocessing phase, the image (Fig. 1(a)) is transformed to facilitate the processing in the next stages. The techniques that have been used are binary thresholding (Fig. 1(b)) and morphological filtering (dilation) (Fig. 1(c)).

As segmentation techniques, the system uses a Canny filter and contour extraction (Fig. 1(d)). The next step is to extract the characteristics of each region:

- *Ratio*: number of pixels in the perimeter.
- *Centroid*: coordinates of the center of gravity.
- *Radio*: centroid distance to the center of the image.
- *Azimuth*: orientation of an object in the image with respect to the  $x$  axis.

If a light is pointing directly to the camera, then the acquired image will be saturated (Fig. 2(a)). In such cases, a big blob can be detected and the image has to be processed again using a higher threshold (Fig. 2(b)). This situation can be very frequent when lights are quite close to the camera.



(a) Saturated image.

(b) Postprocessed image.

Fig. 2. Postprocessing phase.

##### B. Inverse Camera Model

The camera model describes how a 3D scene is transformed into a 2D image. The camera that has been used follows a projection model developed by Pajdla and Bakstein [22] that indicates how a point ( $B$ ) in a 3D reference system can be transformed to a pixel in a 2D image. The model requires the calculation of two angles. On the one hand,  $\theta$  (Fig. 3) is the angle formed between the optical axis of the camera and the beam. This beam is the line from the origin of coordinates of the camera to point  $B$ . On the other hand,  $\varphi$  is the angle between the  $x$ -axis and the projection of the beam on the  $x$ - $y$  plane. Then, the distance  $r$  (Fig. 3) from the image center ( $u_0, v_0$ ) to the coordinates of point  $B$  in the image ( $u_B, v_B$ ) is estimated as:

$$r = a \cdot \tan \frac{\theta}{b} + c \cdot \sin \frac{\theta}{d}, \quad (2)$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are parameters of the model. This function makes it possible to calculate the coordinates of the point in the image ( $u_B, v_B$ ) depending on the azimuth ( $\varphi$ ) and the elevation ( $\theta$ ) (Fig. 3):

$$\left. \begin{aligned} u_B &= u_0 + r \cdot \cos \varphi \\ v_B &= v_0 + r \cdot \sin \varphi \end{aligned} \right\} \quad (3)$$

where  $\beta$  is the ratio between the width and the height of a pixel. These equations (Eqs. 2 and 3) define the omnidirectional camera model.

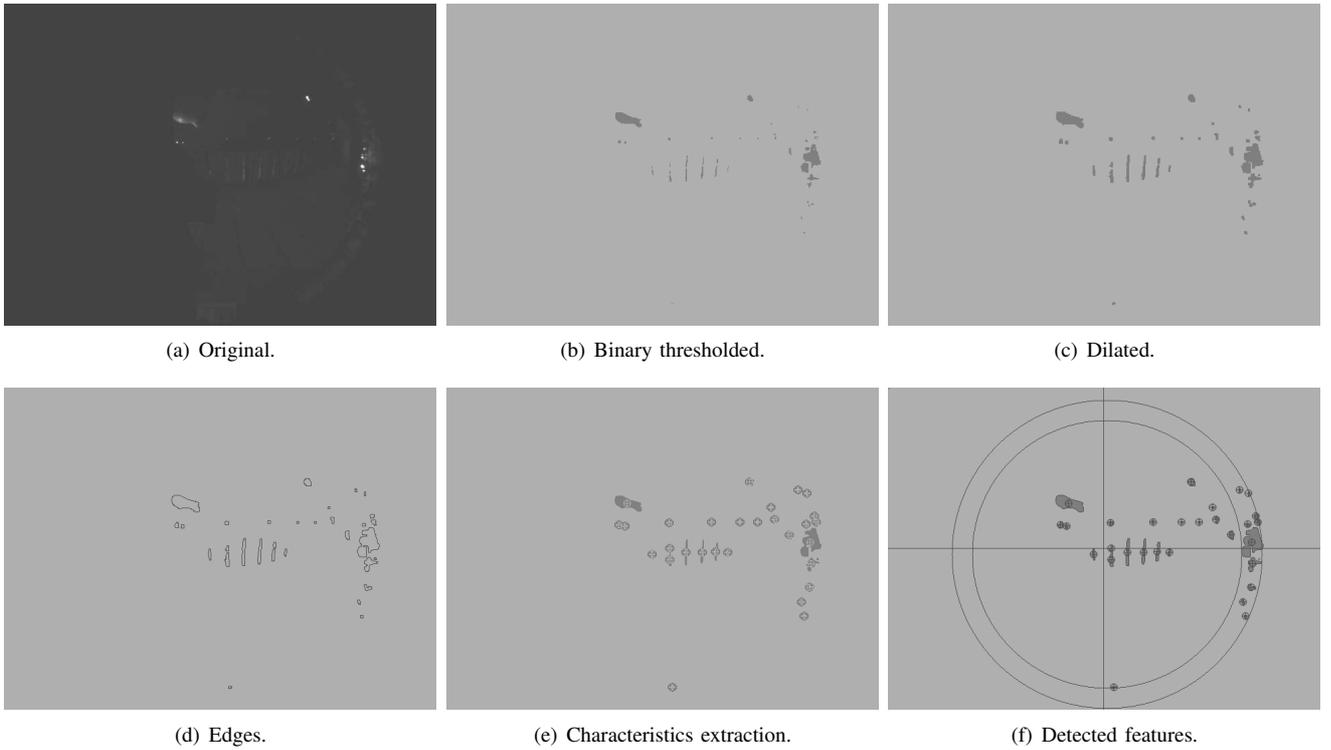


Fig. 1. Feature extraction from an omnidirectional image.

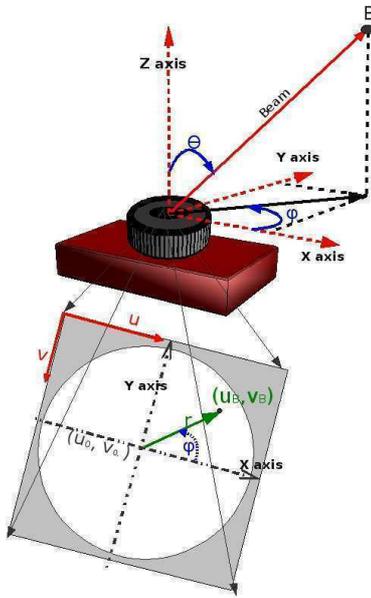


Fig. 3. Projection of a 3D point  $B$  on the image reference system using the omnidirectional camera model.

However, the output of the feature extraction process is a list of pixel coordinates  $((u_l, v_l)$  for the  $l$ -th feature) representing the centroid of each of the possible landmarks. This feature list must be transformed into the measurements list  $(z_t)$ , where each measurement is given by:

$$z_{l,t} = (r_{l,t}, \varphi_{l,t}, \theta_{l,t})^T \quad (4)$$

This transformation requires the inverse camera model, i.e., given a pixel returns the coordinates of the 3D point in the world. This cannot be done due to two limitations:

- The camera model is not invertible.
- The camera is a bearing-only sensor, thus from one image only  $\varphi$  and  $\theta$  could be calculated for a pixel, but not  $r$ .

The first limitation has been solved through a look-up table. Given the coordinates of a pixel, the look-up table provides the values of  $\varphi$  and  $\theta$ . 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  $\varphi$  and  $\theta$  with precisions  $\delta_\varphi$  and  $\delta_\theta$ . Use equations 2 and 3 to obtain the corresponding pixel coordinates.
- 2) Store, for each pixel, the maximum and minimum values of  $\varphi$  and  $\theta$ , as different values could correspond to the same pixel.

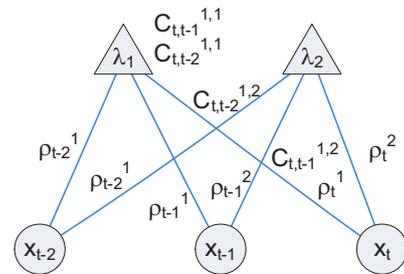


Fig. 4. Determination of the 3D positions of the landmarks.

The second limitation requires the detection of a landmark from two different poses of the robot to get its 3D position. The process to obtain the 3D positions of the landmarks is as follows (Fig. 4):

- 1) Use the Hungarian algorithm to associate the features detected at time  $t$  with the features detected at time  $t - 1$ .
- 2) From each pose of the robot ( $x_t$ ), obtain the 3D line between the robot and the landmark  $\lambda_m$ :  $\rho_t^m$ .
- 3) Let  $P_t$  be the set of lines from  $x_t$  to the landmarks detected at  $t$ , and let  $P_{t-1}$  be the set of lines from  $x_{t-1}$  to the landmarks detected at  $t - 1$ . Calculate the set of intersection points between each line  $\rho_t^m$  and  $\rho_{t-1}^{\hat{c}(m)}$ , where  $\hat{c}(m)$  is the feature in  $t - 1$  associated to feature  $m$  detected in  $t$ .
- 4) Lines in 3D generally don't intersect at a point, but most often only their projection onto a plane intersect. One way to solve the intersection between lines is to calculate the shortest line segment and take its middle point.

## V. RESULTS

The SLAM algorithm has been tested with a *Pioneer 3-DX* robot equipped with an omnidirectional color digital camera (MDCS2) with a fish-eye lens (FE185CO46HA-1, FOV 185°) and a passband infrared filter (IRP) type HOYA RT-830. The camera was placed 1.8m over the floor, in order to minimize the occlusions due to people. Experiments were carried out in an exposition hall of the Domus Museum (A Coruña, Spain). The environment has a size of  $27 \times 7m^2$  and a very uneven floor. The images acquired during localization had a frequency of one second, and the ground truth that has been used to estimate the localization error was obtained using three laser scanners placed in the environment. Three reflective marks were placed on the robot to calculate its position and orientation using the external lasers.

All the experiments have been executed with the following values for the parameters of the algorithm:  $M = 1000$ ,  $p_0 = 0.15$ , and  $p_n = 1$ . The landmarks used for these experiments consisted of a set of lights placed on the ceiling of the environment. Fig. 5 shows the trajectory of the robot using the odometry, the ground truth, and the path estimated by the SLAM algorithm. In order to evaluate this results, the difficulties of the environment have to be taken into account. First, the floor is really uneven, generating a continuous swinging of the camera<sup>1</sup>. This makes measurements very noisy and, therefore, the data association becomes really complex, affecting the positioning of the landmarks. Moreover, as the camera is a bearing-only sensor, the 3D coordinates of each landmark need to be calculated from two consecutive images. This, also introduces noise in the measurements, particularly for landmarks placed far away from the robot.

Although these difficulties, results show a good estimation of the real path of the robot. The errors in the estimated

<sup>1</sup>The associated video can be downloaded from <http://www.gsi.dec.usc.es/mucientes/videos/slam.taros09.wmv>

pose of the robot, both in position and angle, are shown in Fig. 6. The highest error in position is reached during the 180° turning, as the data association becomes really complex due to the imprecision in the measurements. However, after the turning the robot recovers from that failure, approaching again to the real path.

## VI. CONCLUSIONS

A SLAM algorithm, based on FastSLAM, using omnivision has been presented. The main novelties of the approach are the data association process and the detection of the features using omnivision. The algorithm has shown a great robustness, although the difficult conditions of the environment due to noisy measurements, and the limitations of a bearing-only sensor. Experimental results are promising, as the trajectory approaches the real one, and the detected landmarks are placed around their real position. As future work, we plan to improve the data association, use a variable number of particles, and take into account loop closing.

## VII. ACKNOWLEDGMENTS

This work was supported by the Spanish Government and the Galician regional government under grants TIN2005-03844, TIN2008-04008/TSI and INCITE08PXIB262202PR. M. Mucientes is supported by the *Ramón y Cajal* program of the Spanish Ministry of Science and Innovation.

## REFERENCES

- [1] M. Montemerlo and S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Springer-Verlag, 2007.
- [2] C. Gamallo, P. Quintia, C. Regueiro, and M. Mucientes, "Montecarlo localization for a guide mobile robot in a crowded environment based on omnivision," in *Towards Autonomous Robotic Systems*, 2008.
- [3] —, "Localization through omnivision for a tour-guide robot," 2009, vol. 3, no. 1, pp. 25–34, *Journal of Physical Agents*.
- [4] 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.
- [5] —, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico: IJCAI, 2003.
- [6] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.
- [7] J. Civera, A. Davison, and J. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2778–2783.
- [8] E. Eade and T. Drummond, "Scalable monocular SLAM," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 469–476, 2006.
- [9] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, 2008.
- [10] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, 2002.
- [11] S. Se, D. G. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, vol. 21,8, pp. 735–758, 2002. [Online]. Available: <http://www.cs.ubc.ca/spider/lowe/pubs.html>
- [12] R. Sim, P. Elinas, and J. Little, "A study of the rao-blackwellised particle filter for efficient and accurate vision-based SLAM," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 303–318, 2007.

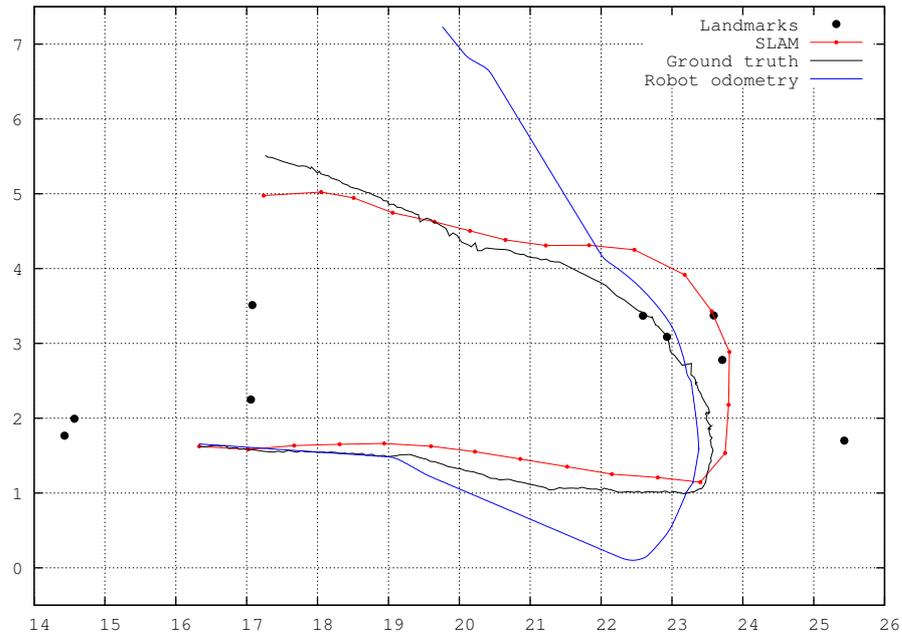


Fig. 5. Path and map obtained by the SLAM algorithm in one of the experiments.

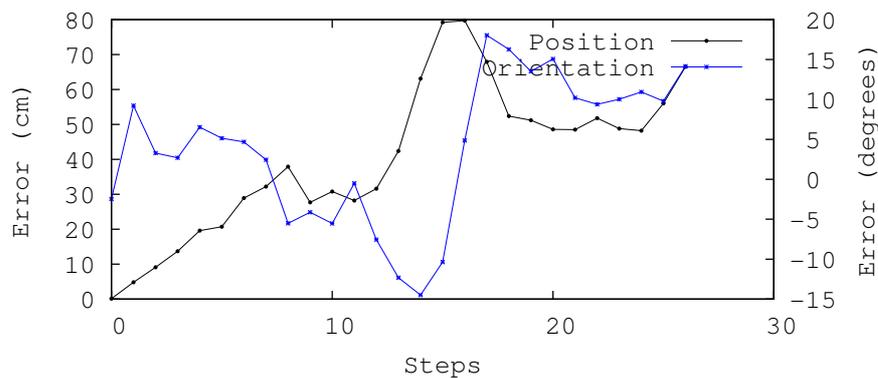


Fig. 6. Errors in the estimated pose of the robot along the path shown in Fig. 5.

- [13] J. Saez, F. Escolano, and A. Penalver, "First steps towards stereo-based 6 DOF SLAM for the visually impaired," in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. (CVPR05)*, 2005, p. 23.
- [14] L. Paz, P. Pinies, J. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 946–957, 2008.
- [15] J. Sola, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multicamera SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 958–968, 2008.
- [16] T. Lemaire and S. Lacroix, "Slam with panoramic vision," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 91–111, Jan.-Feb. 2007.
- [17] P. E. Rybski, S. Roumeliotis, M. Gini, and N. Papanikopoulos, "Appearance-based mapping using minimalistic sensor models," *Autonomous Robots*, vol. 24, no. 3, pp. 229–246, 2008.
- [18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. The MIT Press, 2005.
- [19] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics*, vol. 52, pp. 7–21, 2005.
- [20] C. Gamallo, C. Regueiro, P. Quintía, and M. Mucientes, "Monte Carlo localization for a guide mobile robot in a crowded environment based on omnivision," in *Proceedings of TAROS, Edinburgh (UK)*, 2008, pp. 1–8.
- [21] C. Gamallo, P. Quintía, C. Regueiro, and M. Mucientes, "Localization through omnivision for a tour-guide robot," *Journal of Physical Agents*, vol. 3, pp. 25–34, 2009.
- [22] 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.