Omnidirectional visual SLAM under severe occlusions☆

C. Gamallo[a], M. Mucientes[a], C.V. Regueiro[b]

[a]*Centro de Investigación en Tecnoloxías da Información (CITIUS), Universidade de Santiago de Compostela (Spain).*
[b]*Department of Electronics and Systems, University of A Coruña (Spain).*

Abstract

SLAM (Simultaneous Localization and Mapping) under severe occlusions in crowded environments poses challenges both from the standpoint of the sensor and the SLAM algorithm. In several approaches, the sensor is a camera pointing to the ceiling to detect the lights. Nevertheless, in these conditions the density of landmarks is usually low, and the use of omnidirectional cameras plays an important role due to its wide field of view. On the other hand, the SLAM algorithm has to be also more involved as data association becomes harder in these environments and, also, due to the combination of omnidirectional vision and the characteristics of the landmarks (ceiling lights): conventional feature descriptors are not appropriate, the sensor is bearing-only, measurements are noisier, and severe occlusions are frequent. In this paper we propose a SLAM algorithm (OV-FastSLAM) for omnivision cameras operating under severe occlusions. The algorithm uses a new hierarchical data association method that keeps multiple associations per particle. We have tested OV-FastSLAM in two real indoor environments and with different degrees of occlusion. Also, we have compared the performance of the algorithm with several methods for the data association. Non-parametric statistical tests highlight that OV-FastSLAM has a statistically significant difference, showing a better performance under different occlusion conditions.

Keywords: Hierarchical data association, Severe occlusions, Omnidirectional cameras, Bearing-only visual SLAM

1. Introduction

The operation of a robot during Simultaneous Localization and Mapping (SLAM) in crowded indoor environments introduces a number of difficulties, both from the standpoint of the sensor and the SLAM algorithm. In first place, the sensor pose plays an important role: all the sensors that get data of the plane in which the robot moves should be discarded, as they do not provide useful information when the robot is completely surrounded by people. Many approaches to SLAM in dynamic environments rely on different tracking techniques to disregard moving objects [1, 2]. However, this solution is not valid when the density of moving objects is very high, and the sensor data is totally affected by the presence of people.

Several authors have used cameras pointing at the ceiling [3, 4, 5, 6, 7], as in highly populated environments this is usually the unique people-free area. In particular, most of these approaches use the lights on the ceiling as landmarks, because the ceiling usually has a low number of distinctive objects to map. Also for this reason, SIFT or other types of feature descriptors are ineffective, as the ceiling has a low number of objects from which to extract meaningful descriptors and, very often, these relevant objects are similar or equal.

Conventional cameras are not appropriate in environments with a low density of landmarks, as they detect few of them in each image. Omnidirectional cameras are the most adequate sensor in these conditions, as the number of features in each image is greater and, therefore, the robot has more valuable information to correct its pose and also the pose of the landmarks. The combination of omnidirectional vision and the ceiling lights as features allows to localize the robot and map the environment in crowded environments, although the system must still be able to operate under severe occlu-

☆This work was supported in part by the Spanish Ministry of Economy and Competitiveness under grants TIN2011-22935 and TIN2012-32262, the Galician Ministry of Education under the project EM2014/012, and by the European Regional Development Fund (ERDF/FEDER) under the projects CN2011/058, CN2012/151 and GRC2013/055 of the Galician Ministry of Education.

Email addresses: cristina.gamallo@usc.es (C. Gamallo), manuel.mucientes@usc.es (M. Mucientes), cvazquez@udc.es (C.V. Regueiro)

sions when people is very close to the robot, or due to the point of view of the camera for oriented lights, etc.

We have already mentioned that a SLAM algorithm for crowded environments is also more involved. The cause is that, for the combination of omnidirectional vision and ceiling lights, data association becomes harder due to the following reasons:

- The use of feature descriptors (SIFT, SURF, etc.) facilitates the data association step, as it depends on both the similarity of the feature descriptors of measurements and landmarks and, also, on the geometric distance between them. However, when the landmarks are specific and very similar objects (e.g. the lights), feature descriptors play no role —landmarks are indistinguishable among each other— and the data association has less information.
- Bearing-only sensors, like omnidirectional cameras, make data association more complex because:
  - Distant landmarks may generate very similar measurements for some poses of the robot, although they could be very far apart in the 3D map.
  - For bearing-only SLAM algorithms the pose of recently initialized landmarks is noisier and data association has to cope with a higher uncertainty.
- Measurements coming from lights are noisier because: i) landmarks are on the ceiling and are always detected at distances over several meters —in our tests, at distances typically over 6 m—; ii) the features appear as blobs in the image, with different sizes and shapes that change depending on the point of view and the lightning conditions; iii) small vibrations of the camera generate large errors in the angles of the detected landmarks.
- Severe occlusions are frequent, hindering data association and, also, delaying the reduction of uncertainty in the position of recently initialized landmarks.

In this paper, we present OV-FastSLAM, a SLAM algorithm for omnivision cameras operating under severe occlusions. The proposal is based on the well-known FastSLAM 2.0 approach [8], and has a new data association algorithm that is global and keeps multiple hypothesis per particle. With global methods, we mean that the data association algorithm takes into account all the measurements and landmarks to calculate the likelihood of each complete possible association —the Hungarian method is global— in contrast to local methods that iterate for all the measurements, calculating the likelihood of association for each of them individually —maximum likelihood (ML) is a local approach. The contributions of this paper are:

- Hierarchical data association. We propose a new data association algorithm that is able to cope with all the complexity of omnidirectional vision under severe occlusions. The method is global and manages multiple hypothesis per particle. Also, it is hierarchical, as the association is divided in two stages —landmarks with and without a 3D position— to prioritize the landmarks with a 3D position. The association of the landmarks with 3D position is based on Murty's algorithm [9], that obtains the n-best assignments in polynomial time. On the other hand, the association of landmarks without 3D position uses the Hungarian algorithm [10] to obtain the best assignment.
- Measurements likelihood. We present a method for estimating the likelihood between a measurement and a landmark without 3D position. The method uses all the measurements associated so far to the landmark, picks the 3D position that maximizes the likelihood of all the measurements, and finally selects the minimum probability over all the measurements for the selected 3D position.
- A deep experimental study on the influence of occlusions, in which we analyze the loss in performance under severe occlusions of different degrees. We have compared the results of OV-FastSLAM and other methods for the data association using non-parametric statistical tests.

2. Related work

In the last years many visual SLAM algorithms have been proposed [11], and most of them can be grouped into two categories: those based on filtering techniques, and the approaches that rely on keyframes. We focus the discussion of the related work on those SLAM approaches that use omnidirectional cameras, or the landmarks are ceiling features, or deal with occlusions.

Omnidirectional cameras are cameras with a wide field of view, due to a fish-eye lens or through standard cameras with mirrors (catadioptric cameras). Their use

Table 1: Summary of the related Work.

Reference	**Algorithm**	**Association**	**Camera**	**Landmarks**	**Occ.**
Roda et al. [7]	Information Theory	Mutual Information	Fish-eye (100°)	Ceiling mosaics	—
Lemaire and Lacroix [13]	EKF	Tracking	Catadioptric	Harris corners	—
Andreasson et al. [16, 17]	Graph optimization	ML	Catadioptric	SIFT	—
Rybski et al. [14]	IEKF & linearized ML	Kanade-Lucas-Tomasi	Catadioptric	Featured images	—
Kim and Oh [15]	EKF	ML	Catadioptric + 2D laser	Lines	—
Wongphati et al. [18]	FastSLAM	ML	Catadioptric	Lines	—
Saedan et al. [19]	Particle Filter + image database	ML	Fish-eye	Wavelet decomposition	—
Schlegel and Hochdorfer [12]	EKF	Mahalanobis + geometric	Catadioptric	SIFT	—
Scaramuzza et al. [20]	Visual Odometry + Bag-of-Words	ML + RANSAC	Catadioptric	SIFT	—
Valiente García et al. [21]	Visual Odometry	Global search of similarity	Catadioptric	Fourier signature + SURF	—
Kawewong et al. [22]	Bag-of-Words	ML	Catadioptric	PIRF	YES
Lui and Jarvis [23]	Probabilistic framework	Euclidean distance	Stereo Catadioptric	Haar coefficients	—
Kang et al. [6]	EKF, FastSLAM, NeoSLAM	Nearest neighbor clustering	Standard	Ceiling lights	—
Jeong and Lee [3]	EKF	ML	Fish-eye (150°)	Harris corners	—
Choi et al. [4]	EKF	ML	Standard	Lines	—
Choi et al. [5]	EKF	Lucas-Kanade	Standard	Harris corners + Lines	—

in visual SLAM is receiving an increasing attention, as they cover wider regions of the environment in each image. Roda et al. [7] use this type of camera to build a map of the ceiling from a set of rectified images. The estimation stage is done maximizing the mutual information between two consecutive views, while they construct the ceiling mosaic through an energy minimization algorithm. Moreover, the deviation of the global trajectory is corrected minimizing the entropy of the whole map.

One of the most popular techniques for SLAM with omnidirectional cameras is the Extended Kalman Filter (EKF). For example, Schlegel and Hochdorfer [12] used a panoramic (catadioptric) camera with an EKF algorithm for monocular visual SLAM through SIFT features, and they carried out experiments in largely varying lighting conditions. The approach of Lemaire and Lacroix [13] is also based on EKF-SLAM. They proposed a delayed landmark initialization method that approximates the depth of the position of the landmark through a sum of Gaussians. Loop closure is managed through the comparison between the current image and a database of previous images that are close to the current position. Rybski et al. [14] present two SLAM approaches: online SLAM with an Iterated EKF, and an offline SLAM based on a batch-processed linearized maximum likelihood estimator. They build a topological map from panoramic images, which are stored if they differ from the images in the database according to the Kanade-Lucas-Tomasi algorithm. Finally, Kim and Oh [15] extract the lines of the environment and apply EKF-SLAM, although the 3D position of the lines is obtained in combination with a laser sensor.

Also, there are approaches for omnidirectional cameras based on particle filters. In [18], Wongphati et al. use FastSLAM 1.0, and the landmarks are the vertical lines of the environment. The proposal of Saedan et al. [19] is based on a particle filter, and the map is both topological and metric. Each node of the map stores its pose and an image, and the particle filter estimates the pose of the robot from the nearest node. The loop closure assigns particles to the different nodes according to the similarity between images. Andreasson et al. [16] present a graph-based approach, where the nodes are the poses from which the images where taken, and the arcs represent the relationship between poses through odometry and the similarity between images. They apply the multilevel relaxation algorithm [24] to estimate the global map and the poses through the maximization of the likelihood of all the measurements. In [17], the same authors extend the proposal to multi-robot SLAM.

On the other hand, Scaramuzza et al. [20] employ an omnidirectional camera to create a database of visual words that can be used to identify and close loops. For motion estimation, they use visual odometry based on SIFT features. In the same line, Valiente et al. [21] use a catadioptric camera and visual odometry. Their main contribution was to simplify the matching between features in two consecutive images through the restriction of their poses. Finally, the proposal of Lui and Jarvis [23] relies on a stereo omnidirectional (catadioptric) camera with variable baseline, so it is possible to calculate the 3D position of a feature with only one acquisition. They obtained both a topological map and a 2D grid map, and tests were done both indoors and outdoors.

There are several papers that use ceiling features as landmarks for SLAM, and most of them with standard cameras. The first proposal was an EKF [3], that can be executed in real-time in a very small area. In [4, 5] they used a modified Monte-Carlo algorithm for localization and a standard EKF for SLAM. The landmarks were the ceiling boundaries, the ceiling lights, and circles, but it was tested only for short paths. Finally, in [6] three SLAM algorithms (EKF, FastSLAM and NeoSLAM) were compared on several indoor environments and over short trajectories with just one loop.

The presence of moving objects (dynamic environments) is the main cause of occlusions in SLAM. Several authors have tackled SLAM with detection and tracking of moving objects (DATMO) [25, 26], although both the density of moving objects and the degree of occlusion are usually low. Nevertheless, our focus is not on SLAM and DATMO, but on SLAM approaches under severe occlusions —independently of the source of the occlusion. We are interested in SLAM proposals that explicitly handle occlusions and, also, on those papers that evaluate their approach under varying degrees of occlusion but do not implement any specific occlusion handling —our proposal belongs to the second category. We have found only one approach in this category: the PIRF-Nav 2.0 SLAM algorithm [22, 27]. It uses the PIRF feature detector, which is extremely robust against dynamic changes. The proposal was tested in several environments, including a very challenging environment: a university canteen during lunch time. Authors found that more than 50% of the descriptors (features) came from dynamic objects, which negatively affects the performance and robustness, achieving a recall rate of 88% in the crowded environment.

Table 1 summarizes the main characteristics of the related work.

3. Camera model and landmarks

3.1. Inverse camera model

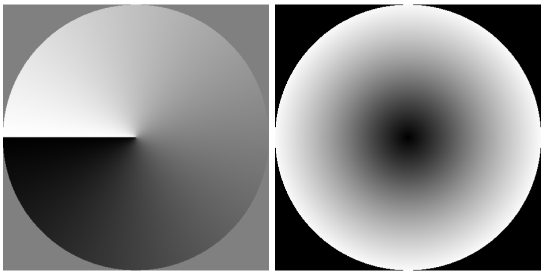
Features —lights on the ceiling— are extracted from the omnidirectional images with the detection process presented in [28]. The output of the feature extraction process is a list of pixel coordinates (u_l, v_l), that represent the centroid of each feature l. This list must be transformed into a measurements list (Z_t), where each measurement $(z_{t,l})$ is given by the azimuth and elevation angles $(\varphi_{t,l},\ \theta_{t,l})$

The camera follows a projection model developed by Pajdla and Bakstein [29] that indicates how a 3D point can be transformed to a pixel in a 2D image. The model is described as a function of the two aforementioned angles φ and θ:

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

where a, b, c, d are parameters of the model, $(u_0,\ v_0)$ are the coordinates of the pixel in 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 —a 3D line for bearing-only sensors. 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 is generated off-line as follows:



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

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

1. Sample the values of φ and θ with precisions δ_φ and δ_θ. Eq. (1) is 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.

3.2. *Measurement noise model*

The noise covariance matrix of the l-th measurement is defined as:

$$Q_l = \begin{pmatrix} Q_l^\varphi & 0 \\ 0 & Q_l^\theta \end{pmatrix} = \begin{pmatrix} \left(\frac{\Delta_{np}}{r}\right)^2 & 0 \\ 0 & \left(\frac{\pi\Delta_{np}}{2r_{max}}\right)^2 \end{pmatrix} \qquad (2)$$

where Δ_{np} represents the uncertainty —in pixels— of the features extraction process, and r and r_{max} are respectively the distances —in pixels— from the center of the image to the feature and to the border of the image (Fig. 2). Q_l^φ is obtained as the squared angle of an arc with length Δ_{np} and radius r. Therefore, Q_l^φ depends on θ (Eq. 1). Fig. 3 shows this dependency: the maximum error is at the center of the image, as at that point we do not have any information on the value of φ; on the other hand, at the border of the image, errors in the position of the features have little influence on φ. Finally, Q_l^θ is defined as the squared angle that corresponds to a segment of length Δ_{np} along a radius of the image.

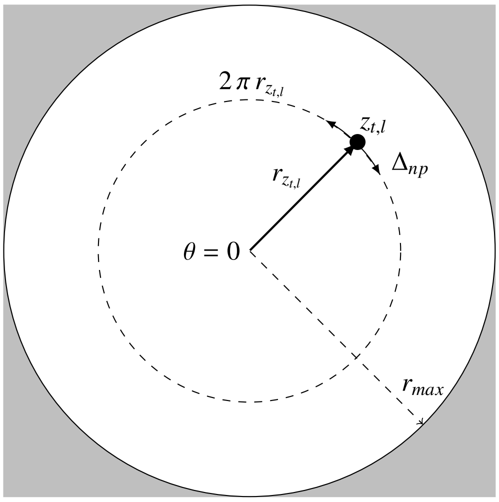
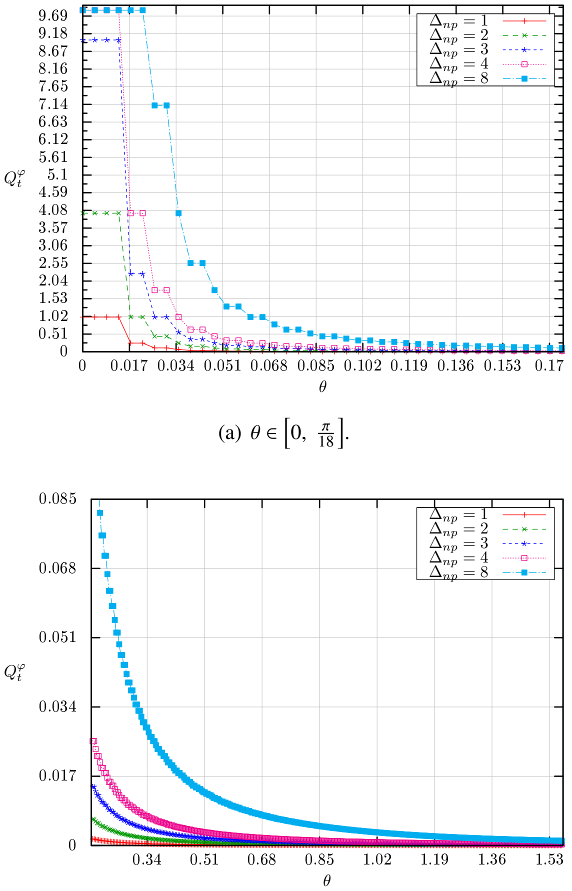


Figure 2: Estimation of Q_l.

3.3. *Map*

The feature-based map is composed of two types of landmarks: (i) those with 3D positions, called landmarks; (ii) and those without a 3D position, called candidate landmarks to distinguish them from the landmarks with 3D position (Fig. 4).



(a) $\theta \in \left[0,\ \frac{\pi}{18}\right]$.

(b) $\theta \in \left[\frac{\pi}{18},\ \frac{\pi}{2}\right]$.

Figure 3: Q_l^φ as a function of θ for different values of Δ_{np}.

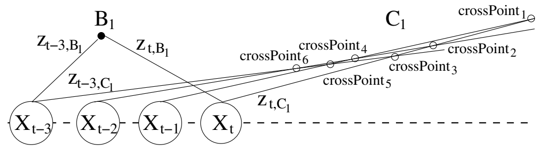


Figure 4: A typical example of a landmark (B_1) and a candidate landmark (C_1).

Algorithm 1 OV-FastSLAM (Z_t, u_t, Y_{t-1})

```
1:  for $k = 1$ to $M$ do
2:      for $s = 1$ to $N_\Psi$ do
3:          Get particle $k$ with data association $s$ from $Y_{t-1}$: $x^{k,s}_{t-1}$, $\{B^{k,s}_{t-1,1}, \ldots, B^{k,s}_{t-1,N^{k,s}_{t-1}}\}$, $\{C^{k,s}_{t-1,1}, \ldots, C^{k,s}_{t-1,\eta^{k,s}_{t-1}}\}$
4:          $\widehat{x_t} = g\left(x^{k,s}_{t-1}, u_t\right)$
5:          $\Phi^{k,s}_{t,1} = measurementsLikelihood\_1\,()$
6:      end for
7:      $\{\Psi^{k,1}_{t,1}, \ldots, \Psi^{k,N_\Psi}_{t,1}\} = dataAssociation\_1\,()$
8:      for $s = 1$ to $N_\Psi$ do
9:          $\Phi^{k,s}_{t,2} = measurementsLikelihood\_2\,()$
10:         $\Psi^{k,s}_{t,2} = dataAssociation\_2\,()$
11:         $robotPoseUpdate\,()$
12:         $mapUpdate\,()$
13:     end for
14: end for
15: $Y_t = resample\,()$
```

- A landmark $B^{k,s}_{t,j}$ —j is the landmark index, k is the particle, s the data association hypothesis, and t the timestamp— contains the information of the 3D position, represented by a Gaussian distribution of mean $\mu^{k,s}_{t,j}$ and covariance $\Sigma^{k,s}_{t,j}$, and the number of times that has been detected, $i^{k,s}_{t,j}$.

- Candidate Landmarks $C^{k,s}_{t,j}$ contain a set of measurements $Z^{k,s}_{t,j} = \{z_{t-\tau,l_{t-\tau}}, \ldots, z_{t,l_t}\}$ that are used to calculate a set of possible 3D positions, each one with an associated probability. Over time, as new measurements are associated to the landmark, the set of candidate 3D positions is modified and the probabilities evolve until the correct —and most probable— position is assigned to the candidate landmark.

4. OV-FastSLAM algorithm

OV-FastSLAM is based on the FastSLAM 2.0 algorithm [8], which uses a Rao-Blackwellized particle filter [30] to represent the posterior probability distribution. The robot path is estimated with a particle filter, and each particle contains both a robot pose and a feature-based map. The features of the map are represented by Gaussian distributions and they are updated with EKFs. The inputs to OV-FastSLAM (Alg. 1) are the set of measurements at time t (Z_t), the control (u_t), and the previous set of particles (Y_{t-1}). Each particle k contains N_Ψ data associations to cope with all the complexity of omnidirectional vision under severe occlusions. For each association s there is an estimated robot pose, $x^{k,s}_{t-1}$, and a map of $N^{k,s}_{t-1}$ landmarks, $\{B^{k,s}_{t-1,1}, \ldots, B^{k,s}_{t-1,N^{k,s}_{t-1}}\}$, and $\eta^{k,s}_{t-1}$ candidate landmarks $\{C^{k,s}_{t-1,1}, \ldots, C^{k,s}_{t-1,\eta^{k,s}_{t-1}}\}$.

OV-FastSLAM iterates over the M particles and the N_Ψ associations to obtain the weights ($w^{k,s}$). Data association is hierarchical, i.e., it is divided in two stages —landmarks and candidate landmarks association— to prioritize landmarks association, as they are more reliable than candidate landmarks. Lines 2-6 calculate the measurements likelihoods for the first level of the data association, which is solved in line 7. The second loop over the N_Ψ associations (lines 8-13) calculates the measurements likelihoods and the data association of the second level, the robot pose update and the map update.

Finally, OV-FastSLAM resamples following the effective sample size criterion [31], calculated as [32]:

$$M_{eff} = \frac{1}{\sum\limits_{i=1}^{M} (\widetilde{w}^k)^2} \qquad (3)$$

where $\widetilde{w}^k$ is the normalized particle weight. When all the particles have very similar weights, M_{eff} takes its maximum value, indicating that the target proposal distribution is correctly approximated. However, when the variance on the particles weights increases, the value of M_{eff} decreases, reflecting a poor approximation of the distribution. OV-FastSLAM resamples whenever $M_{eff} < M/2$ [32]. Only the best association of each particle passes to the resampling stage (Sec. 4.2.3).

FastSLAM 2.0 tracks several hypothesis (one per particle), but each particle only keeps its best data association. Whenever an occlusion or a complex data association occurs, it is not unusual that most of the particles make wrong data associations. On the other hand, OV-FastSLAM keeps several data associations per particle, which makes highly probable to track the correct hypothesis. Moreover, as the decision on which is the most probable data association is delayed until resampling takes place, the data association of OV-FastSLAM is more robust to noisy measurements, severe occlusions, and the complexity of bearing-only sensors.

Algorithm 2 OV-FastSLAM: *measurementsLikelihood_1* ()

```
for $j = 1$ to $N^{k,s}_{t-1}$ do
    $\bar{z}_j = h\left(\mu^{k,s}_{t-1,j},\ \widehat{x_t}\right)$
    $H_{x,j} = \nabla_{x_t} h\left(\mu^{k,s}_{t-1,j},\ \widehat{x_t}\right)$
    $H_m = \nabla_m h\left(\mu^{k,s}_{t-1,j},\ \widehat{x_t}\right)$
    for $l = 1$ to $N_{Z_t}$ do
        $Q_{j,l} = Q_l + H_m \Sigma^{k,s}_{t-1,j} H_m^T$
        $\Sigma_x = [H_{x,j}^T Q_{j,l}^{-1} H_{x,j} + R_t^{-1}]^{-1}$

        $\mu_x = \Sigma_x H_{x,j}^T Q_{j,l}^{-1}\left(z_{t,l} - \bar{z}_j\right) + \widehat{x_t}$
        $\widehat{z} = h\left(\mu^{k,s}_{t-1,j},\ \mu_x\right)$
        $\phi_{j,l} = (2\pi)^{-\frac{Dim\left(Q_{j,l}\right)}{2}} \left|Q_{j,l}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_{t,l} - \widehat{z})^T Q_{j,l}^{-1}(z_{t,l} - \widehat{z})\right\}$
    end for
end for
return $\Phi^{k,s}_{t,1} = \left[\phi_{j,l}\right]$
```

4.1. Measurements likelihood

The likelihood of association ($\phi_{j,l}$) between a landmark j and a measurement l is obtained by Alg. 2. It follows the FastSLAM 2.0 approach, which models the probability of association by a Gaussian with mean equal to the measurement prediction, and with a covariance that depends on the landmark covariance ($\Sigma^{k,s}_{t-1,j}$) and the measurement noise (Q_l).

The probability that a measurement is associated to candidate landmark j is calculated in Alg. 3. The process is also based on a Gaussian probability distribution but, as candidate landmarks do not have a position — but a set of possible positions—, the process is more involved. The calculation of $\phi_{j,l'}$ has the following steps:

Algorithm 3 OV-FastSLAM: *measurementsLikelihood_2* ()

```
 1: for $j = 1$ to $\eta^{k,s}_{t-1}$ do
 2:     for $l' = 1$ to $N^{k,s}_{Z'_t}$ do
 3:         $\phi_{max} = 0$
 4:         $validCross = false$
 5:         for $i = 1$ to $|Z^{k,s}_{t-1,j}|$ do
 6:             $x_c = crossPoint(z_{t,l'}, z_i)$
 7:             $\phi_c = 1$
 8:             $\phi_{min} = 1$
 9:             for $q = 1$ to $|Z^{k,s}_{t-1,j}|$ do
10:                 $\bar{z} = h\left(x_c,\ \widehat{x_{t(q)}}\right)$
11:                 $H_m = \nabla_m h\left(x_c,\ \widehat{x_{t(q)}}\right)$
12:                 $Q_{j,l'} = Q_q + H_m \Sigma_0 H_m^T$
13:                 $\phi_q = (2\pi)^{-\frac{Dim\left(Q_{j,l'}\right)}{2}} \left|Q_{j,l'}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(z_q - \bar{z}\right)^T Q_{j,l'}^{-1}\left(z_q - \bar{z}\right)\right\}$
14:                 $\phi_c := \phi_c \phi_q$
15:                 if $\phi_q < \phi_{min}$ then
16:                     $\phi_{min} = \phi_q$
17:                 end if
18:             end for                              ▷ For $q = 1$ to $|Z^{k,s}_{t-1,j}|$
19:             if $\left(\widehat{z_{t,l'}, z_i} > \gamma_{min}\ \vee\ !validCross\right) \wedge \phi_c > \phi_{max}$ then
20:                 $\phi_{max} = \phi_c$
21:                 $\phi_{j,l'} = \phi_{min}$
22:                 if $\widehat{z_{t,l'}, z_i} > \gamma_{min}\ \wedge\ !validCross$ then
23:                     $validCross = true$
24:                 end if
25:             end if
26:         end for                                  ▷ For $i = 1$ to $|Z^{k,s}_{t-1,j}|$
27:     end for                                      ▷ For $l' = 1$ to $N^{k,s}_{Z'_t}$
28: end for                                          ▷ For $j = 1$ to $N^{k,s}_{t-1}$
29: return $\Phi^{k,s}_{t,2} = \left[\phi_{j,l'}\right]^T$
```

1. For each measurement $z_i \in Z^{k,s}_{t-1,j}$ —the set of measurements belonging to $C^{k,s}_{t-1,j}$—, we calculate the crosspoint between z_i and $z_{t,l'}$ (Alg. 3, line 6), with $l' = 1, \ldots, N^{k,s}_{Z'_t}$, being $N^{k,s}_{Z'_t}$ the number of measurements at time t not associated in the first level of the data association process.
   The probability assigned to the crosspoint (ϕ_c) is the joint probability of all the measurements in $Z^{k,s}_{t-1,j}$ for the 3D position (x_c) of the crosspoint (lines 9-18). For each measurement z_q, the probability that it was generated from x_c (ϕ_q) is modeled by a Gaussian with mean equal to the measurement prediction. This prediction uses the crosspoint position and the prediction of the pose of the robot at $t\,(q)$, which is the timestamp of measurement q. Moreover, the covariance depends on the crosspoint covariance (Σ_0, the initial covariance of a landmark) and the measurement noise (Q_q).
2. Select the crosspoint that maximizes the probability over all the measurements (ϕ_c) in $Z^{k,s}_{t-1,j}$. The angle between the measurements that define the crosspoint must be over a threshold γ_{min} (line 19) to have a high confidence in its 3D position.
3. $\phi_{j,l'}$ will be the minimum over all the measurements for the selected crosspoint (line 21). This condition is very restrictive, and helps to eliminate candidate landmarks with associated measurements that do not have a good matching with the

most probable crosspoint.

4.2. *Hierarchical data association*

Landmarks association is based on Murty's algorithm [9], that obtains the n-best assignments in polynomial time. This algorithm has been modified by Cox and Miller [33] in order to solve multiple assignment problems at the same time, and also to change the termination condition, as hypothesis with a likelihood below a certain percentage of the best hypothesis can be discarded. All those measurements that are not associated in the first stage, pass to the second stage —candidate landmarks association— which uses the Hungarian algorithm [10] to obtain the best assignment. The Hungarian method is a combinatorial optimization algorithm that solves assignment problems in polynomial time.

Both Murty's and Hungarian algorithms use a cost matrix (Φ), where each element $\phi_{a,b}$ represents the probability of association[1] between a and b. The Hungarian method returns an ambiguity matrix (Ψ) —Murty's algorithm returns a set of n ambiguity matrices—, where each element $\psi_{a,b}$ takes a value of 1 or 0, indicating the association or not of a with b. Moreover, Ψ has to fulfill two conditions:

$$\sum_b \psi_{a,b} = 1,\ \forall a \text{ and } \qquad \sum_a \psi_{a,b} \in \{0,\ 1\},\ \forall b \tag{4}$$

The first condition indicates that each row must have exactly one assignment, while the second equation reflects that each column must have one or zero associations.

4.2.1. *First level of the data association*

It is solved with Murty's algorithm using the cost matrix shown in Fig. 5. $\Phi^{k,s}_{t,1}$ was generated by Alg. 2, but at this stage it is extended to form an $N^{k,s}_{t-1} \times \left(N_{Z_t} + N^{k,s}_{t-1}\right)$ matrix with a row for each landmark.

The left side of the matrix is obtained from Alg. 2. There is a column for each measurement, and each element $\phi_{j,l}$ represents the probability of association of landmark j with measurement l.

The right side is a diagonal submatrix, with a column per landmark. This gives the chance to make no assignment to a landmark. The probability of no association is defined as:

$$\phi_{na,j} = \phi_{new,j}\phi_{out,j} \tag{5}$$

The first one is the probability that none of the measurements comes from landmark j. We define the probability that a measurement comes from a new landmark as:

$$\phi_{new,l} = (2\pi)^{-\frac{Dim(Q_l)}{2}} |Q_l|^{-\frac{1}{2}} \exp\{-\frac{1}{2}\xi^2_{new}\} \tag{6}$$

where Q_l is the measurement noise (Eq. 2) and ξ_{new} is a parameter that represents the difference (error) —in number of standard deviations— between the measurement and the expected measurement. Values over ξ_{new} indicate that the most probable association is that measurement l does not come from a mapped landmark. However, we want to model that none of the measurements comes from landmark j ($\phi_{new,j}$), and we estimate this probability replacing in Eq. 6 measurement l ($z_{t,l}$) with $\bar{z}_j$, which is the predicted measurement for the landmark. This only modifies the value of Q_l (Eq. 2) and, in this way, we take into account the expected measurement noise instead of the noise of an specific measurement.

The second probability in Eq. 5 models the likelihood that the landmark is outside the field of view of the camera:

$$\phi_{out,j} = \begin{cases} \text{if } \theta <= \theta_{max} & 0 \\ \\ \text{if } \theta > \theta_{max} & \\ \qquad 1 - \exp\{-\frac{1}{2}\frac{\left(\theta_{\bar{z}_j} - \theta_{max}\right)^2}{Q^\theta_j}\} & \end{cases} \tag{7}$$

where $\theta_{\bar{z}_j}$ is the elevation angle for the estimated measurement of landmark j, θ_{max} is a parameter, and Q^θ_j is Q^θ_l (Eq. 2) for $\bar{z}_j$. This probability modulates $\phi_{new,j}$: if the landmark is outside the field of view ($\phi_{out,j} = 1$), the probability that the landmark is not associated is $\phi_{new,j}$, while if the landmark is inside the field of view its probability of no association is very low.

With this definition for $\Phi^{k,s}_{t,1}$, an association has to be selected for each landmark (this includes the possibility that the landmark is not detected). On the other hand, some measurements could be not assigned and, therefore, they will pass to the second association level.

4.2.2. *Second level of the data association*

It is based on the Hungarian method and uses the cost matrix described in Fig. 6. $\Phi^{k,s}_{t,2}$ was calculated with Alg. 3 and, at this stage, it is extended to create a $N^{k,s}_{Z'_t} \times \left(\eta^{k,s}_{t-1} + N^{k,s}_{Z'_t}\right)$ matrix with a row for each measurement ($z_{t,l'}$) that was not associated in the first level. Therefore, each measurement that passed to the second level has to be associated.

[1] As it is a cost matrix, each element is actually $-\log \phi_{a,b}$, but to keep notation simple we will use $\phi_{a,b}$.

$$\begin{array}{cc}
 & \begin{array}{ccc|cccc} z_{t,1} & \longleftrightarrow & z_{t,N_{Z_t}} & B^{k,s}_{t-1,1} & \longleftrightarrow & & B^{k,s}_{t-1,N^{k,s}_{t-1}} \end{array} \\
\begin{array}{c} B^{k,s}_{t-1,1} \\ \updownarrow \\ \updownarrow \\ B^{k,s}_{t-1,N^{k,s}_{t-1}} \end{array} & \left(\begin{array}{ccc|cccc}
\phi_{1,1} & \cdots & \phi_{1,N_{Z_t}} & \phi_{na,1} & 0 & \cdots & 0 \\
\phi_{2,1} & \cdots & \phi_{2,N_{Z_t}} & 0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\
\phi_{N^{k,s}_{t-1},1} & \cdots & \phi_{N^{k,s}_{t-1},N_{Z_t}} & 0 & \cdots & 0 & \phi_{na,N^{k,s}_{t-1}}
\end{array}\right)
\end{array}$$

Figure 5: Cost matrix for the first level of the data association, $\Phi^{k,s}_{t,1}$.

$$\begin{array}{cc}
 & \begin{array}{ccc|cccc} C^{k,s}_{t-1,1} & \longleftrightarrow & C^{k,s}_{t-1,\eta^{k,s}_{t-1}} & C_{new,1} & \longleftrightarrow & & C_{new,N^{k,s}_{Z'_t}} \end{array} \\
\begin{array}{c} z_{t,1'} \\ \updownarrow \\ \updownarrow \\ z_{t,N^{k,s}_{Z'_t}} \end{array} & \left(\begin{array}{ccc|cccc}
\phi_{1,1} & \cdots & \phi_{1,\eta^{k,s}_{t-1}} & \phi_{new,1} & 0 & \cdots & 0 \\
\phi_{2,1} & \cdots & \phi_{2,\eta^{k,s}_{t-1}} & 0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\
\phi_{N^{k,s}_{Z'_t},1} & \cdots & \phi_{N^{k,s}_{Z'_t},\eta^{k,s}_{t-1}} & 0 & \cdots & 0 & \phi_{new,N^{k,s}_{Z'_t}}
\end{array}\right)
\end{array}$$

Figure 6: Cost matrix for the second level of the data association, $\Phi^{k,s}_{t,2}$.

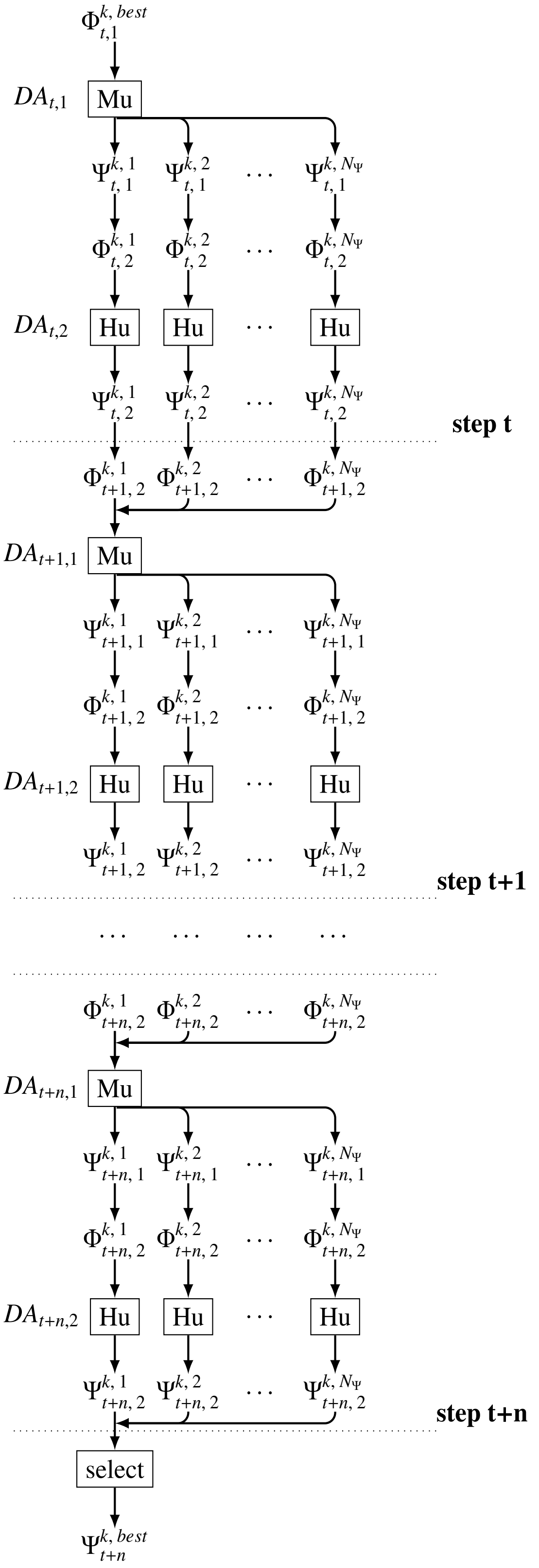
The left side is the matrix generated by Alg. 3. There is a column per candidate landmark, and each element $\phi_{j,l'}$ represents the probability of association of candidate landmark j with measurement l' —the matrix with elements $\phi_{j,l'}$ is transposed (Alg. 3, line 29) to generate $\Phi^{k,s}_{t,2}$.

The right side is a diagonal matrix with a column per measurement that passed to the second level. $\phi_{new,l'}$ represents the probability that the measurement comes from a new candidate landmark (Eq. 6). In this association level, we do not include the probability that the candidate landmark is outside the field of view, as the positions of the candidate landmarks undergo rapid changes.

4.2.3. *Full data association process*

Fig. 7 shows the full data association process for a particle. An association cycle comprises several iterations of OV-FastSLAM, starting after the last resampling at time $t-1$ and ending before the next resampling at time $t+n$. At the beginning of iteration t, there is a unique cost matrix per particle $\Phi^{k,best}_{t,1}$ that enters the first level of the data association process (DA^t_1), generating the best N_Ψ associations with Murty's algorithm. From each of these associations ($\Psi^{k,s}_{t,1}$), the algorithm generates a cost matrix for the second level ($\Phi^{k,s}_{t,2}$) and the Hungarian method returns the best association ($\Psi^{k,s}_{t,2}$). Therefore, at the end of the first iteration, each particle has N_Ψ different associations and, consequently, N_Ψ different robot poses and maps.

In the second iteration ($t+1$), the first association level receives N_Ψ cost matrices —each one represents a different association problem— and generates the best N_Ψ assignments altogether. This means that with Murty's algorithm all the cost matrices compete to generate associations that are in the top N_Ψ, i.e., it could happen that part of the cost matrices do not contribute to the top N_Ψ associations. The best associations are again used to generate the cost matrices for the second level ($\Phi^{k,s}_{t+1,2}$), and for each of them the Hungarian method returns the best assignment. The process is repeated until resampling takes place. In the last iteration, the best per particle association $\Psi^{k,best}_{t+n}$ —it represents the two association levels— is selected based on the best $\Psi^{k,s}_{t+n,1}$ and its corresponding $\Psi^{k,s}_{t+n,2}$. Therefore, at the end of the full association cycle and before the resampling step, each par-



$\Phi_{t,1}^{k,best}$
$DA_{t,1}$
Mu
$\Psi_{t,1}^{k,1}$ $\Psi_{t,1}^{k,2}$ … $\Psi_{t,1}^{k,N_\Psi}$
$\Phi_{t,2}^{k,1}$ $\Phi_{t,2}^{k,2}$ … $\Phi_{t,2}^{k,N_\Psi}$
$DA_{t,2}$
Hu Hu … Hu
$\Psi_{t,2}^{k,1}$ $\Psi_{t,2}^{k,2}$ … $\Psi_{t,2}^{k,N_\Psi}$
step t
$\Phi_{t+1,2}^{k,1}$ $\Phi_{t+1,2}^{k,2}$ … $\Phi_{t+1,2}^{k,N_\Psi}$
$DA_{t+1,1}$
Mu
$\Psi_{t+1,1}^{k,1}$ $\Psi_{t+1,1}^{k,2}$ … $\Psi_{t+1,1}^{k,N_\Psi}$
$\Phi_{t+1,2}^{k,1}$ $\Phi_{t+1,2}^{k,2}$ … $\Phi_{t+1,2}^{k,N_\Psi}$
$DA_{t+1,2}$
Hu Hu … Hu
$\Psi_{t+1,2}^{k,1}$ $\Psi_{t+1,2}^{k,2}$ … $\Psi_{t+1,2}^{k,N_\Psi}$
step t+1
… … … …
$\Phi_{t+n,2}^{k,1}$ $\Phi_{t+n,2}^{k,2}$ … $\Phi_{t+n,2}^{k,N_\Psi}$
$DA_{t+n,1}$
Mu
$\Psi_{t+n,1}^{k,1}$ $\Psi_{t+n,1}^{k,2}$ … $\Psi_{t+n,1}^{k,N_\Psi}$
$\Phi_{t+n,2}^{k,1}$ $\Phi_{t+n,2}^{k,2}$ … $\Phi_{t+n,2}^{k,N_\Psi}$
$DA_{t+n,2}$
Hu Hu … Hu
$\Psi_{t+n,2}^{k,1}$ $\Psi_{t+n,2}^{k,2}$ … $\Psi_{t+n,2}^{k,N_\Psi}$
step t+n
select
$\Psi_{t+n}^{k,best}$

Figure 7: A cycle of the full data association process for a particle.

ticle has a single data association —and a single robot pose and map.

4.3. Robot pose update

Algorithm 4 OV-FastSLAM: *robotPoseUpdate* ()

```
if $\sum_{j=1}^{N_{t-1}^{k,s}} \psi_j = 0$ then
    $x_t^{k,s} \sim p\left(x_t | x_{t-1}^{k,s},\ u_t\right)$
else
    $\Sigma_x = R_t$
    $\mu_x = \widehat{x_t}$
    for $j = 1$ to $N_{t-1}^{k,s}$ do
        if $\psi_j > 0 \wedge \Upsilon_j = 1$ then
            $\Sigma_x := [H_{x,j}^T Q_{j,\psi_j}^{-1} H_{x,j} + \Sigma_x^{-1}]^{-1}$
            $\mu_x := \mu_x + \Sigma_x H_{x,j}^T Q_{j,\psi_j}^{-1} \left(z_{t,\psi_j} - \overline{z}_j\right)$
        end if
    end for
    $x_t^{k,s} \sim N(\mu_x,\ \Sigma_x)$
end if
```

Algorithm 4 describes the robot pose update for multiple simultaneous measurements. OV-FastSLAM only takes into account the landmarks with a low uncertainty ($\Upsilon_j = 1$) in this stage. The goal of the loop of Alg. 4 is to build the Gaussian proposal distribution from which the robot pose will be sampled. As the proposal distribution shrinks with each observation, the order in which the landmarks are processed is important: at the beginning, the proposal covariance is higher and, therefore, the corrections proposed by these observations will have a higher influence. OV-FastSLAM processes landmarks in increasing order of $Tr\left(Q_{j,\psi_j}\right)$ —the covariance matrix trace for the measurement associated to landmark j (ψ_j)—, i.e., lower covariances are processed in first place. In this way, both the landmark and the observation covariances are taken into account, and those with a higher confidence will be processed first.

4.4. Map update

As OV-FastSLAM manages different types of landmarks, the map update process is divided in two stages: landmarks and candidate landmarks update.

4.4.1. Landmarks update

Algorithm 5 describes the landmarks update process and the particle weight calculation. The landmarks position update recalculates the predicted measurement (and the matrices that depend on it) as the robot pose has

Algorithm 5 OV-FastSLAM: *mapUpdate* ()

```
1:  $\omega^{k,s} = 1$
2:  for $j = 1$ to $N^{k,s}_{t-1}$ do
3:      if $\psi_j > 0$ then
4:          $i^{k,s}_{t,j} = i^{k,s}_{t-1,j} + 1$
5:          $\widetilde{z} = h\left(\mu^{k,s}_{t-1,j},\ x^{k,s}_t\right)$
6:          $\widetilde{H}_m = \nabla_m h\left(\mu^{k,s}_{t-1,j},\ x^{k,s}_t\right)$
7:          $\widetilde{Q}_{j,\psi_j} = Q_{\psi_j} + \widetilde{H}_m \Sigma^{k,s}_{t-1,j} \widetilde{H}^T_m$
8:          $K = \Sigma^{k,s}_{t-1,j} \widetilde{H}^T_m \widetilde{Q}^{-1}_{j,\psi_j}$
9:          $\mu^{k,s}_{t,j} = \mu^{k,s}_{t-1,j} + K\left(z_{t,\psi_j} - \widetilde{z}\right)$
10:         $\Sigma^{k,s}_{t,j} = \left(I - K\widetilde{H}_m\right)\Sigma^{k,s}_{t-1,j}$
11:         if $\Upsilon_j = 1$ then
12:             $\widetilde{H}_x = \nabla_{x_t} h\left(\mu^{k,s}_{t-1,j},\ x^{k,s}_t\right)$
13:             $L = \widetilde{H}_x R_t \widetilde{H}^T_x + \widetilde{Q}_{j,\psi_j}$
14:             $\widehat{w} = (2\pi)^{-\frac{Dim(L)}{2}} |L|^{-\frac{1}{2}}$
                        $\exp\left\{-\frac{1}{2}\left(z_{t,\psi_j} - \widetilde{z}\right)^T L^{-1} \left(z_{t,\psi_j} - \widetilde{z}\right)\right\}$
15:         else                                          ▷ If $\Upsilon_j = 1$
16:             $\widehat{w} = 1$
17:         end if                                        ▷ If $\Upsilon_j = 1$
18:     else                                              ▷ If $\psi_j > 0$
19:         $\mu^{k,s}_{t,j} = \mu^{k,s}_{t-1,j}$
20:         $\Sigma^{k,s}_{t,j} = \Sigma^{k,s}_{t-1,j}$
21:         if $\Upsilon_j = 1$ then
22:             $\widehat{w} = \phi_{out,j}$
23:         else                                          ▷ If $\Upsilon_j = 1$
24:             $\widehat{w} = 1$
25:         end if                                        ▷ If $\Upsilon_j = 1$
26:         if $\mu^{k,s}_{t-1,j}$ is inside perceptual range of $x^{k,s}_t$
    then
27:             $i^{k,s}_{t,j} = i^{k,s}_{t-1,j} - 1$
28:         end if                                ▷ If $\mu^{k,s}_{t-1,j}$ is inside
29:     end if                                            ▷ If $\psi_j > 0$
30:     $\omega^{k,s} = \omega^{k,s} \cdot \widehat{w}$
31: end for
```

been modified in the robot pose update stage. Thereafter, the position update follows the same steps as FastSLAM 2.0, i.e., the standard EKF update. Moreover, $i^{k,s}_{t,j}$ counts the number of observations for landmark j, and decreases its value whenever the landmark is not detected but is inside the field of vision of the camera (Alg. 5, lines 26-27).

The particle-association weight ($\omega^{k,s}$) is the product of the partial weights corresponding to each landmark ($\widehat{w}$). OV-FastSLAM only takes into account the landmarks with a low uncertainty ($\Upsilon_j = 1$) for weight calculation. If the landmark has been observed at time t, the partial weight follows a Gaussian distribution with mean equal to the measurement prediction, and the covariance is a combination of the measurement noise, the landmark covariance, and the motion noise. However, if the landmark is not observed, OV-FastSLAM also incorporates this negative evidence (Alg. 5, line 22), setting the partial weight to the probability that the landmark is outside the field of view of the camera.

4.4.2. Candidate landmarks update

Candidate landmarks update is performed taking into account the outcome of the data association:

- Candidate landmark $C^{k,s}_{t,j}$ has an assigned measurement. The observation is added to $Z^{k,s}_{t,j}$, the observation counter is increased, and the set of possible 3D positions and probabilities is updated.
- Candidate landmark $C^{k,s}_{t,j}$ has no assigned measurement. If the landmark is inside the field of view of the camera, the observation counter is decreased: $i^{k,s}_{t,j} = i^{k,s}_{t-1,j} - I_{not}$, where I_{not} is the number of consecutive iterations in which the candidate landmark was not observed. If $i^{k,s}_{t,j}$ is under 0, the candidate landmark is deleted.
- The measurement belongs to a new candidate landmark, $C^{k,s}_{t,j}$. Both the measurements set and the observation counter are initialized.

4.4.3. Transformation from candidate landmarks to landmarks

Alg. 6 transforms candidate landmarks into landmarks, i.e., it calculates the initial position of each landmark as the crosspoint that maximizes the probability of the set of measurements that belong to the candidate landmark. First, for each pair of measurements in $Z^{k,s}_{t,j}$, the algorithm calculates the crosspoint (x_c). Then, it calculates the probability (ϕ_c) of all the measurements for each crosspoint (lines 8-14). For each candidate landmark, the algorithm selects the crosspoint with the maximum probability (lines 15-18). The selected crosspoint also has to fulfill that the baseline between the positions of the camera which generated the crosspoint is over a threshold (γ_{min}). Finally, Alg. 6 returns the initial positions of the landmarks, or *null* if the candidate landmarks were not transformed.

5. Results

OV-FastSLAM has been validated with a *Pioneer 3-DX* equipped with an omnidirectional color digital camera (MDCS2) with a fish-eye lens (FE185CO46HA-1,

Algorithm 6 OV-FastSLAM: *candidateToLandmark*()

```
for $j = 1$ to $\eta_t^{k,s} \wedge \Psi_j > 0$ do
    $\phi_{max} = 0$
    $x^j_{landmark} = null$
    for $l' = 1$ to $|Z^{k,s}_{t,j}|$ do
        for $i = l' + 1$ to $|Z^{k,s}_{t,j}|$ do
            $x_c = crossPoint(z_{l'}, z_i)$
            $\phi_c = 1$
            for $q = 1$ to $|Z^{k,s}_{t,j}|$ do
                $\bar{z} = h\left(x_c, \widehat{x_{t(q)}}\right)$
                $H_m = \nabla_m h\left(x_c, \widehat{x_{t(q)}}\right)$
                $Q_{j,l'} = Q_q + H_m \Sigma_0 H_m^T$
                $\phi_q = (2\pi)^{-\frac{Dim\left(Q_{j,l'}\right)}{2}} \left|Q_{j,l'}\right|^{-\frac{1}{2}}$
                        $\exp\left\{-\frac{1}{2}\left(z_q - \bar{z}\right)^T Q_{j,l'}^{-1}\left(z_q - \bar{z}\right)\right\}$
                $\phi_c := \phi_c \phi_q$
            end for                                ▷ For $q = 1$ to $|Z^{k,s}_{t,j}|$
            if $\phi_c > \phi_{max} \wedge \widehat{z_{t,l'}, z_i} > \gamma_{min}$ then
                $\phi_{max} = \phi_c$
                $x^j_{landmark} = x_c$
            end if
        end for                                    ▷ For $i = 1$ to $|Z^{k,s}_{t,j}|$
    end for                                        ▷ For $l' = 1$ to $|Z^{k,s}_{t,j}|$
end for                                            ▷ For $j = 1$ to $\eta_t^{k,s} \wedge \Psi_j > 0$
return $\{x^1_{landmark}, \ldots, x^{\eta_t^{k,s}}_{landmark}\}$
```

FOV 130°) and a band-pass infrared filter HOYA RT-830, in two different indoor environments: a sports hall (*Pío XII*) and a museum (*Domus*). Table 2 summarizes the main characteristics of the environments and tests. Due to the height of the ceilings, the landmarks are typically detected at distances over 6 m; for instance, in *Pío XII*, landmarks were detected at distances up to 20 meters.

We have tested OV-FastSLAM under very different lighting conditions with great results; the influence of the lighting conditions is low due to the use of the infrared filter. In fact, in environment *Pío XII*, the sports hall has large windows through which natural light enters. This creates glare and reflections in the ceiling, which can be managed by OV-FastSLAM. Reflections influence OV-FastSLAM in two ways: hiding lights or creating false lights. When a reflection hides a light, this is equivalent to an occlusion, and OV-FastSLAM can cope with these situations (Sec. 5.1). On the other hand, when reflections create false lights, they could be considered as candidate landmarks. However, they are never transformed into landmarks, as this process is very restrictive and requires that all the associated measurements verify the initial position of the landmark with a high probability. Finally, and when no new measurements are associated to the candidate landmark, it is deleted from the map.

In all the experiments the parameters of OV-FastSLAM took the following values: $a = 406.1510$, $b = 2.9951$, $c = 2.0066$, $d = 0.2079$, $\beta = 1.0$ (camera model parameters, Eq. 1), $\Delta_{np} = 2$ (*Pío XII*), 4 (*Domus*) (measurement noise model, Eq. 2), $\gamma_{min} = 1.22\,rad$ (7°) (Alg. 3), $\xi_{new} = 8$ (Eq. 6), $\theta_{max} = 1.26\,rad$ (72°) (Eq. 7), $\Sigma_0 = [0.0025\ 0.0; 0.0\ 0.0025]$. In order to validate our proposal we have used the following SLAM algorithms:

- OV-FastSLAM(M, N_Ψ). OV-FastSLAM with M particles and the described hierarchical data association: the first level uses Murty's algorithm with N_Ψ associations, and the second level is based on the Hungarian method.
- H-H(M). OV-FastSLAM with M particles but with a hierarchical data association in which both levels use the Hungarian algorithm.
- Also, we have tried FastSLAM 1.0, FastSLAM 2.0, and OV-FastSLAM with two non hierarchical data associations based on Murty's algorithm and on the Hungarian method. In all the cases, the SLAM algorithms were unable to close the loops and the errors in the path and the map were unaffordable.

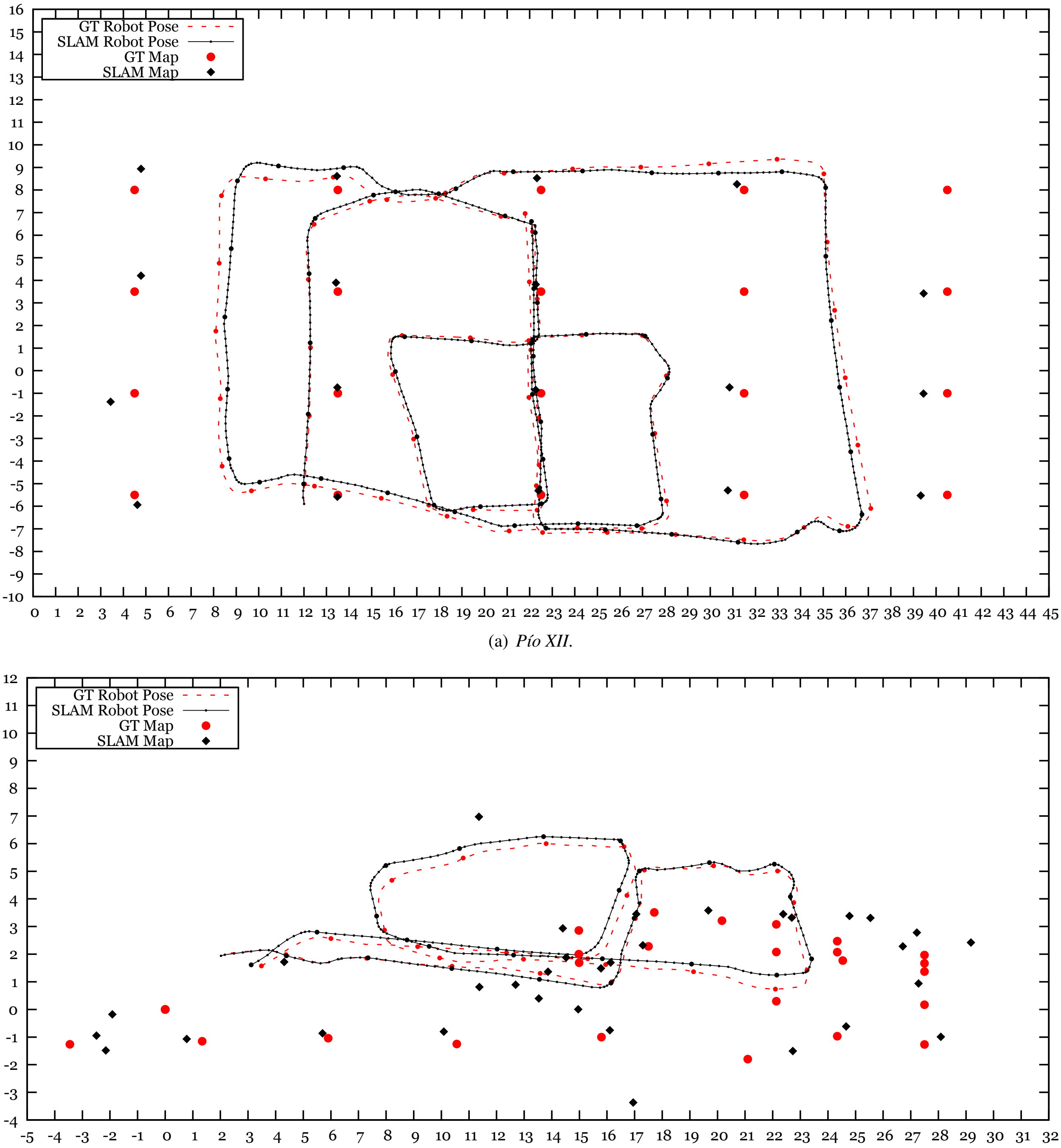
Fig. 8 shows the estimated trajectory, the obtained map and, also, the real trajectory (*GT robot*) and the real map (*GT map*) of the two test environments for OV-FastSLAM(5, 2)[2,3]. The algorithm was able to close the loops in all the situations. Fig. 9 illustrates the average error in the position of the robot along the path for different values of $M \cdot N_\Psi$, as the product of these parameters is the number of hypothesis tracked by OV-FastSLAM(M, N_Ψ). H-H(M) has also been included for comparison purposes. OV-FastSLAM was run for different values in the number of associations (N_Ψ). As the SLAM algorithm is stochastic, each value is the average over the two environments of the mean over 10 runs with different seeds (for each environment), to highlight the reliability of the algorithm independently of the randomness due to the sampling steps.

[2] The associated videos can be downloaded from: `http://persoal.citius.usc.es/manuel.mucientes/videos/OV-FastSLAM_pio.mp4` and `OV-FastSLAM_domus.mp4`.

[3] The algorithm was run on an Intel(R) Core(TM) i5-2500 3.30GHz CPU at an average frequency of 2.63 Hz.

Table 2: Characteristics of the environments.

Env.	Size	Distance	Loops	#Img.	Img. Freq.	v_{max}	ω_{max}	#Land.	Land. height
DOMUS	$27x7\,m^2$	$72\,m$	4	548	$2\,Hz$	$0.36\,m/s$	$0.58\,rad/s$	36	$11.30\,m,\ 3.25\,m$
Pío XII	$24x24\,m^2$	$174\,m$	6	1180	$2\,Hz$	$0.36\,m/s$	$0.58\,rad/s$	20	$6.5\,m$



(a) *Pío XII.*

(b) *Domus*

Figure 8: Most probable trajectories and maps in the two test environments for OV-FastSLAM(5, 2).

Table 3: Non-parametric test for the performance with different values of $M \cdot N_\Psi$ (Fig. 9) with $\alpha = 0.05$.

i	Alg.	Ranking	z	p	α/i	Hypothesis
—	OV-FastSLAM(M, 5)	1.6	—	—	—	—
2	H-H(M)	2.7	2.46	0.014	0.025	Rejected
1	OV-FastSLAM(M, 2)	1.7	0.22	0.823	0.050	Accepted

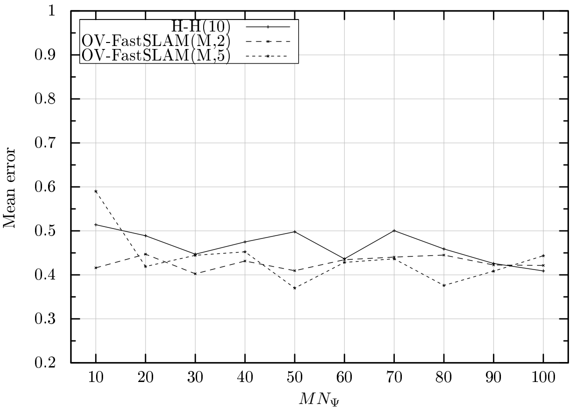
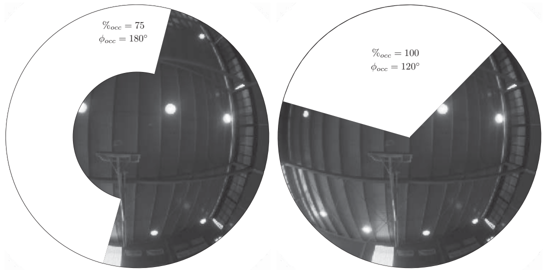


Figure 9: Average performance of the algorithms in the two test environments with different values of $M \cdot N_\Psi$.

We started the comparison in $M \cdot N_\Psi = 10$ to let OV-FastSLAM(M, 5) have at least two particles. As can be seen, both OV-FastSLAM(M, N_Ψ) consistently outperform H-H(M) except when the number of particles is very low (OV-FastSLAM(2, 5) vs. H-H(10)), or when the number of particles increases over a threshold (approximately 100 particles for these test environments). In this last case, the performance of all the algorithms becomes very similar.

We have compared the data association methods using non-parametric statistical tests [34, 35]. We first applied the Friedman test that computes the ranking of the results of the algorithms, and rejects the null hypothesis —which states that the results of all the algorithms are equivalent— with a given confidence —significance level (α). In second place, we applied Holm's post-hoc test for detecting significant differences among the results. The tests were performed for the error in position for the different values of $M \cdot N_\Psi$, i.e., we have always compared algorithms with the same number of tracked hypothesis. Table 3 summarizes the tests results using OV-FastSLAM(M, 5) as the control algorithm. The ranking column was generated by the Friedman test, and taking into account these values, the i was assigned. z is the value calculated by Holm's test, and p is its corresponding p-value. All the hypothesis with $p < \alpha/i$ are rejected, which means that the algorithms are different with a confidence level of α. The test shows that the difference in performance between both OV-FastSLAM(M, N_Ψ) and H-H(M) association is statistically significant. On the other hand, although OV-FastSLAM(M, 5) is better than OV-FastSLAM(M, 2), the test cannot reject the null hypothesis.



(a) $\phi_{occ} = 180°$, $\%_{occ} = 75$. (b) $\phi_{occ} = 120°$, $\%_{occ} = 100$.

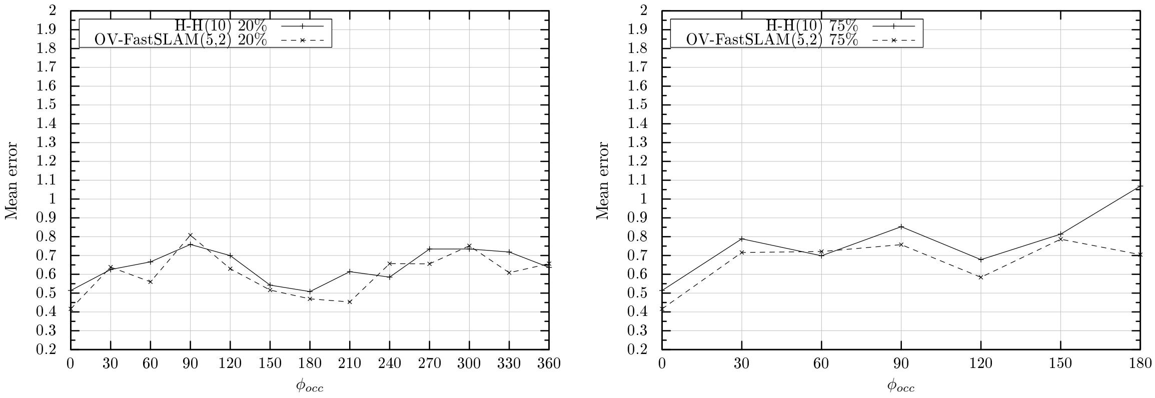
Figure 10: Typical occlusion masks.

Table 4: Non-parametric test for the performance under occlusions.

Alg.	Ranking
OV-FastSLAM(5, 2)	1.4
H-H(10)	1.6
Friedman p-value = 0.054	

5.1. Occlusions

OV-FastSLAM was designed to operate under severe occlusions. In order to test the influence of the occlusions in the performance of the algorithm, we have evaluated OV-FastSLAM under continuous occlusions of different degrees. The occlusions were artificially generated by superimposing a mask on the images. In this way, it is possible to measure the loss in performance for different degrees of occlusion. The masks are built as the intersection of a circular sector with an annulus (Fig. 10). Thus, they are defined with two parameters: $\phi_{occ} \in [0°, 360°]$ and $\%_{occ} \in [0, 100]$. ϕ_{occ} represents the angle of the circular sector, and $\%_{occ}$ the percentage of the area of the image corresponding to the annulus. The placement of the mask on the image —the angle of rotation of the mask with respect to the image— is randomly modified at each time instant. In this way we are



(a) $\%_{occ}$ = 20.

(b) $\%_{occ}$ = 75.

Figure 11: Performance of the algorithms for different degrees of occlusion in the *Pío XII* environment.

able to simulate occlusions due to people moving, non mapped objects, etc. in a realistic way.

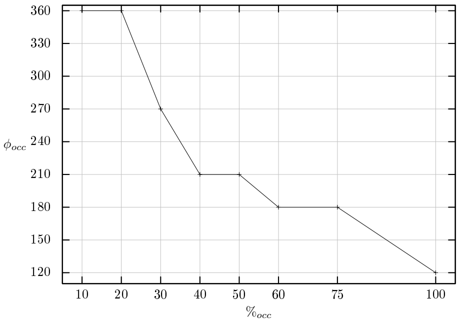


Figure 12: Maximum values of ϕ_{occ} for which the performance does not degrade significantly.

The experiments took place in the two test environments, for values of ϕ_{occ} each 30° and for 8 values of $\%_{occ}$: $\{10-60,\ 75,\ 100\}$. OV-FastSLAM was run with $M = 5$ and $N_\Psi = 2$, and it was compared with H-H(10). Table 4 summarizes the non-parametric Friedman test —Holm test was not necessary as we only compared two algorithms. The test points out that both algorithms are different with a confidence level of $\alpha = 0.06$ and, as the ranking of OV-FastSLAM(5, 2) is better, we can conclude that OV-FastSLAM(5, 2) outperforms H-H(10) in the occlusions test.

Fig. 11 depicts the performance of both algorithms for two values of $\%_{occ}$ and the whole range of ϕ_{occ}. Those experiments in which the error in both algorithms is over a threshold (two times the minimum error without occlusion) have been discarded, as we consider that both algorithms failed. For $\%_{occ} = 20$ both algorithms are able to operate under occlusions (up to $\phi_{occ} = 360°$) with an acceptable performance, although OV-FastSLAM(5, 2) is in most of the cases better. Although at a first glance, one should hope that the error increases with the degree of occlusion, the analysis must be quite more subtle. For example, when two landmarks are close, the occlusion of one of them can generate a failure in the data association —the occluded landmark is associated with the feature corresponding to the visible landmark. This happens for example at $\phi_{occ} = 90°$, but when the degree of occlusion increases (and both landmarks become occluded) the error diminishes again. These errors in the data association are reflected in the performance of the algorithms due to their influence in the initialization of the landmarks, the elimination of landmarks, or the correction of the position of the robot with the information of the detected landmarks.

For $\%_{occ} = 75$ (severe occlusion), the pattern is similar, but for $\phi_{occ} \geq 180°$ both algorithms usually fail as most of the landmarks are occluded. Fig. 12 shows the values of ϕ_{occ} for which the performance does not degrade significantly. As expected ϕ_{occ} decreases with the increase in $\%_{occ}$, but even for $\%_{occ} = 100$, OV-FastSLAM is able to get a reasonable error when a third of the image is occluded.

6. Conclusions

We have presented OV-FastSLAM, a SLAM algorithm for omnivision cameras to operate in indoor environments under severe occlusions. The main contributions of the proposal are: i) the hierarchical data association method, which uses Murty's algorithm in the first level and the Hungarian method in the second one; ii) the measurements likelihood for the landmarks without 3D position; and iii) a deep experimental study on the influence of severe occlusions on the performance of OV-FastSLAM.

OV-FastSLAM has been validated in two real and complex environments with different degrees of occlusion showing a good performance. Moreover, we have also compared OV-FastSLAM with different data association methods. Results of the non-parametric tests reflect a statistically significant difference between the algorithms, and highlight the lower error of OV-FastSLAM and its ability to operate under severe occlusions.

[1] D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, D. Sorrenti, Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments, in: ICRA International Workshop on Safe navigation in open and dynamic environments: Application to autonomous vehicles, 2009.

[2] J. Solà, Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach, Ph.D. thesis, Institut National Polytechnique de Toulouse-INPT (2007).

[3] W. Jeong, K. Lee, CV-SLAM: A new ceiling vision-based SLAM technique, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005, pp. 3195–3200.

[4] H. Choi, D. Kim, J. Hwang, E. Kim, Y. Kim, CV-SLAM using ceiling boundary, in: IEEE Conference on Industrial Electronics and Applications (ICIEA), 2010, pp. 228–233.

[5] H. Choi, S. Jo, E. Kim, CV-SLAM using line and point features, in: International Conference on Control, Automation and Systems (ICCAS), 2012, pp. 1465–1468.

[6] J. Kang, S. Kim, S. An, S. Oh, A new approach to simultaneous localization and map building with implicit model learning using neuro evolutionary optimization, Applied Intelligence (2012) 1–28.

[7] J. Roda, J. Sáez, F. Escolano, Ceiling mosaics through information-based SLAM, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego (USA), 2007, pp. 3898–3904.

[8] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, in: Proceedings of the AAAI National Conference on Artificial Intelligence, 2002.

[9] K. Murty, An algorithm for ranking all the assignments in order of increasing cost, Operations Research 16 (1968) 682–687.

[10] H. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistics Quarterly 2 (1-2) (1955) 83–97.

[11] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J. Rendón-Mancha, Visual simultaneous localization and mapping: a survey, Artificial Intelligence Review (2012) 1–27.

[12] C. Schlegel, S. Hochdorfer, Advances in Service Robotics, In-Tech, 2008, Ch. Localization and mapping for service robots: Bearing-only SLAM with an omnicam.

[13] T. Lemaire, S. Lacroix, SLAM with panoramic vision, Journal of Field Robotics 24 (1-2) (2007) 91–111.

[14] P. Rybski, S. Roumeliotis, M. Gini, N. Papanikopoulos, Appearance-based mapping using minimalistic sensor models, Autonomous Robots 24 (3) (2008) 229–246.

[15] S. Kim, S. Oh, SLAM in indoor environments using omnidirectional vertical and horizontal line features, Journal of Intelligent and Robotic Systems 51 (1) (2008) 31–43.

[16] H. Andreasson, T. Duckett, A. Lilienthal, Mini-SLAM: Minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity, in: IEEE International Conference on Robotics and Automation (ICRA), 2007, pp. 4096–4101.

[17] H. Andreasson, T. Duckett, A. Lilienthal, A minimalistic approach to appearance-based visual SLAM, IEEE Transactions on Robotics 24 (5) (2008) 991–1001.

[18] M. Wongphati, N. Niparnan, A. Sudsang, Bearing only FastSLAM using vertical line information from an omnidirectional camera, in: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009, pp. 1188–1193.

[19] M. Saedan, C. Lim, M. Ang, Appearance-based SLAM with map loop closing using an omnidirectional camera, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2007, pp. 1–6.

[20] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees, Robotics and Autonomous Systems 58 (6) (2010) 820–827.

[21] D. Valiente García, L. Fernández Rojo, A. Gil Aparicio, L. Payá Castelló, O. Reinoso García, Visual odometry through appearance-and feature-based method with omnidirectional images, Journal of Robotics 2012.

[22] A. Kawewong, N. Tongprasit, O. Hasegawa, PIRF-Nav 2.0: Fast and online incremental appearance-based loop-closure detection in an indoor environment, Robotics and Autonomous Systems 59 (10) (2011) 727 – 739.

[23] W. Lui, R. Jarvis, A pure vision-based topological SLAM system, The International Journal of Robotics Research 31 (4) (2012) 403–428.

[24] U. Frese, P. Larsson, T. Duckett, A multilevel relaxation algorithm for simultaneous localization and mapping, IEEE Transactions on Robotics 21 (2) (2005) 196–207.

[25] C. Wang, C. Thorpe, S. Thrun, M. Hebert, H. Durrant-Whyte, Simultaneous localization, mapping and moving object tracking, The International Journal of Robotics Research 26 (9) (2007) 889–916.

[26] T. Vu, O. Aycard, N. Appenrodt, Online localization and mapping with moving object tracking in dynamic outdoor environments, in: IEEE Intelligent Vehicles Symposium, 2007, pp. 190–195.

[27] N. Tongprasit, A. Kawewong, O. Hasegawa, PIRF-Nav 2: Speeded-up online and incremental appearance-based SLAM in an indoor environment, in: IEEE Workshop on Applications of Computer Vision (WACV), 2011, pp. 145–152.

[28] C. Gamallo, C. Regueiro, P. Quintía, M. Mucientes, Omnivision-based KLD-Monte Carlo localization, Robotics and Autonomous Systems 58 (2010) 295–305.

[29] H. Bakstein, T. Pajdla, Panoramic mosaicing with a 180° field of view lens, in: Proceedings of the Third Workshop on Omnidirectional Vision, 2002, pp. 60–67.

[30] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, The MIT Press, 2005.

[31] J. Liu, Metropolized independent sampling with comparisons to rejection sampling and importance sampling, Statistics and Computing 6 (2) (1996) 113–119.

[32] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo methods in practice, Statistics for Engineering and Information Science, Springer, 2001.

[33] I. Cox, M. Miller, On finding ranked assignments with application to multitarget tracking and motion correspondence, IEEE Transactions on Aerospace and Electronic Systems 31 (1995)

486–489.

[34] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[35] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.