Monte Carlo localization for a guide mobile robot in a crowded environment based on omnivision

Cristina Gamallo, Carlos V. Regueiro, Pablo Quintía and Manuel Mucientes

***Abstract*— This work presents a localization system for a robot guide on a crowded environment based on omnidirectional vision and a map of ceiling landmarks. The developed approach uses a Monte Carlo particle filter to manage uncertainty, both on observations and control, in order to track the position of the robot. We describe how landmarks are detected on each image and how the problem of landmark association is posed. To demonstrate the robustness and reliability of our system, we present experiments carried out in a real environment, the Domus Museum in A Coruña (Spain). Results show that the proposed localization system can run on real time and along middle-long trajectories.**

I. INTRODUCTION

Robot localization is one of the most important tasks in autonomous mobile robotic. Most of the action a robot has to perform require the knowledge of its position. Determining the location of a mobile robot is estimating the Cartesian coordinates and angular orientation relative to an external reference frame. It requires to be reliable, robust and executable on real time.

Our system has been developed for a guide robot (Fig. 1) at the Domus Museum (Fig. 2 and 3) located in A Coruña (Spain). This environment is highly populated and, therefore, typical range sensors like laser o ultrasonic devices do not work accurately in normal conditions, we are not allowed to modify the environment introducing artificial landmarks to facilitate localization. Thus we have decided to use artificial vision and natural landmarks of the environment to estimate the position of the robot.

A landmark can be any distinctive and recognizable object on the environment. This work uses the lights placed on the ceiling of the museum (Fig. 2) which are easily detectable, repetitive and usually visible along large trajectories. On the other hand, any building has lights, so there is no need for prior preparation of the environment in order to use our localization method. The main problem is the difficulty to distinguish among different landmarks, as they are usually equal.

The camera is pointing to the ceiling and fixed on the robot over 1.5 m above it (Fig. 1), so that their movements are restricted by the degrees of freedom of the robot (a Pioneer IIAT) and the noise or occlusions generated by moving people is minimized. As our camera is an omnidirectional camera (Fig. 4), it means a very wide field of vision (FOV around 185^o) which covers the half space of the environment and obtains a lot of information about it in each acquisition. It is noteworthy that the floor of the environment is very irregular and produces swinging in the camera support. This increases the noise both in the observations (images) and in the estimation of the robot's position with the odometry system.

A Monte Carlo localization algorithm [15] has been used to solve the tracking position problem. The key-point of the

Fig. 1. The vision system in the Guide Mobile Robot, based on Pioneer IIAT, used at the Domus Museum in A Coruña (Spain). The omnivision camera is marked with a circle.

(a) (b)

Fig. 2. Examples of landmarks in the Domus Museum: (a) Some lights labeled with E and A on Fig. 3; (b) Lights labeled with B on Fig. 3.

C. Gamallo and M. Mucientes are within the Intelligent Systems Group, Dept. of Electronics and Computer Science, University of Santiago de Compostela (Spain). cgamallo@usc.es, manuel@dec.usc.es

C.V. Regueiro and P. Quintía is with the Computer Architecture Group, Department of Electronic and Systems, University of A Coruña (Spain). {cvazquez,pquintia}@udc.es

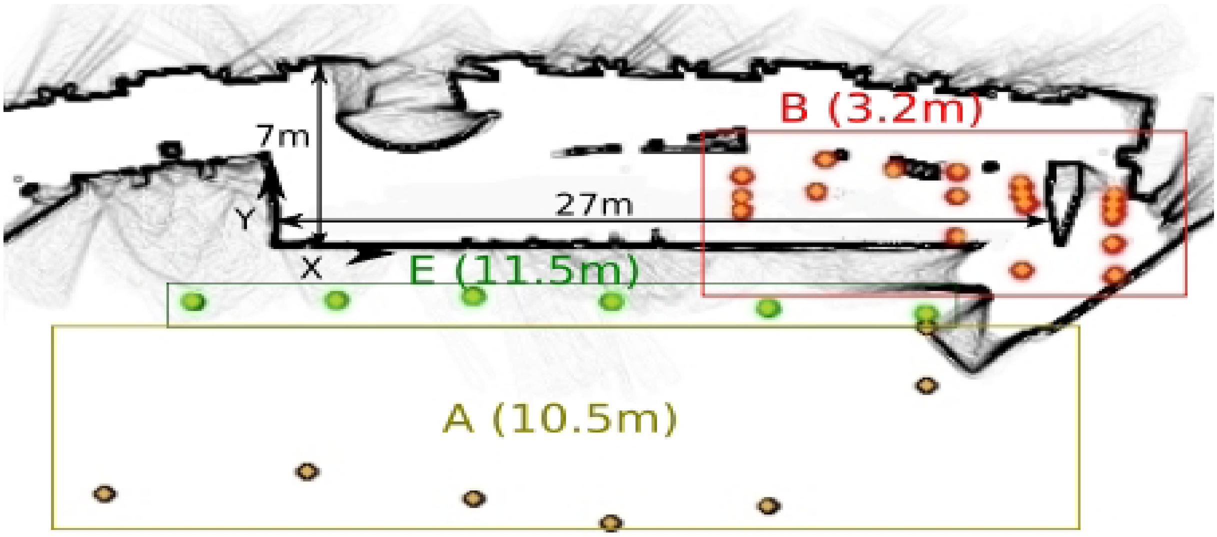


Fig. 3. Map and landmarks (lights) used to localize the guide robot in the Domus Museum. Lights (circles) with the same height (shown in parenthesis) are classified in three regions.

(a) (b)

Fig. 4. (a) Omnidirectional lens (185^o FOV). (b) Omnidirectional image acquired with our omnidirectional camera.

algorithm is to manage the uncertainly in robot perception and action by means of Probability Density Functions (PDF). In the case of Monte Carlo localization, the PDF is estimated with a particle filter.

Section II exposes an overview of related works. The next three sections describe the vision system and landmark detection on the images, the omnidirectional camera and the Monte Carlo localization process. Section VI presents the experimental results in a real environment, and finally, the last section points out conclusions and future work.

II. RELATED WORK

There has been extensive research in the literature to solve the localization problem using vision. Most of them use landmarks in the environment as a reference to obtain the robot positions. The works that use artificial landmarks were not addressed here since they can not be applied in our environment because of its peculiarities, that we have described in the last section.

Concerning omnidirectional vision to locate a mobile robot, the first work was published in 1986 by Cao et at [4]. Although few related studies were published before the end of the nineties. Nowadays such systems, for example [13], [3], [10], [1], [9] and [12], have become popular due to their low cost in addition to the benefit of having a very wide field of vision.

The probabilistic approach is the most used in recent publications. One example is [14]. In which, a Monte Carlo localization algorithm is suggested to solve global localization problem using a camera. They used a visual map of the ceiling, obtained by mosaicing, and localized the robot using a simple scalar brightness measurement as the sensor input. The camera pointed to the ceil just the same settings our system has. But unlike the present work, this system is sensitivity to bumps and as a result of the small FOV of the camera, there are instants that hardly any lights can be seen. This causes more uncertainly in the system. Another similar approach is [12], which use an omnivision camera oriented to the ceiling too, but it is based on information theory to get the global trajectory. The main problem of this work is the hight cost of computing.

[1] and [10] use Monte Carlo filtered too but they create a database with images of every route and their positions. The robot can be localized by correlation between the captured images and the database images in real time. These systems have the drawback that they can not work in other routes on the environment.

Menegatti et at. [9] have developed a system which uses a chromatic map of the floor to compute the robot pose. They obtained similar results but their system is limited a environments with natural color transitions and it is also light-sensitive.

Another popular probabilistic algorithm is the Kalman filter and it was considered in [7], [6], [8] and [11] to implement SLAM using a single camera. A limitation of this system is it need a large number of distinguishable features to perform accurately. On the other hand the number of features it can handle to process on real-time are restricted. Moreover t Kalman filter is not scalable. So that it is limited to small rooms. In contrast our work can operate in a large hall using only 31 landmarks.

III. VISION SYSTEM

The vision system (Fig.4) consists on a color digital camera MDCS2, equipped with an omnidirectional lens (*fish-eye*) FE185CO46HA-1 and an infrared baseband filter (IRP), model HOYA IR85. The camera is mounted on the robot with its optical axis perpendicular to the plane of the ground and pointing to the ceiling (Fig. 1).

(a) (b)

Fig. 5. Optical filtering effect: (a) original image, (b) filtered image (only IR light is detected).

The use of a high resolution omnidirectional lens, with a very wide field of vision (FOV), reduces the number of landmarks needed, as they can be seen from more different points of the environment 4.

The infrared filter baseband (IRP) attenuates the components of visible light and only lets the close range pass *IR*(Fig. 5).

A. Landmark detection

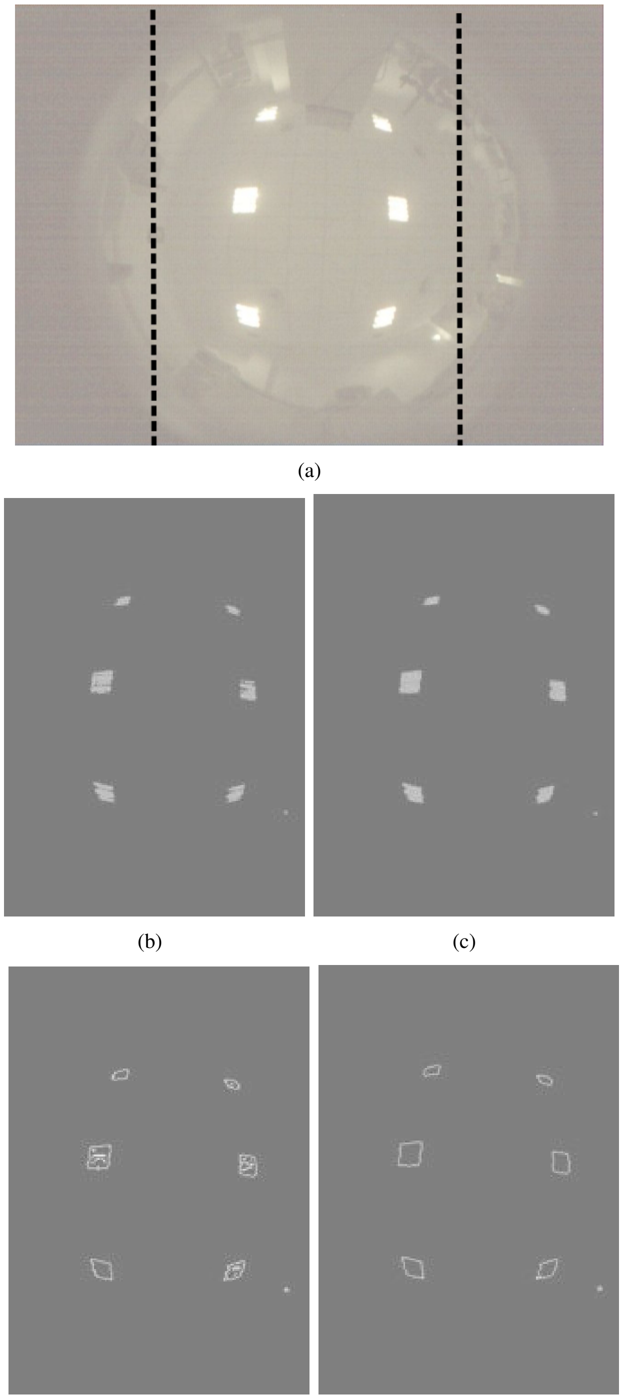
The process of detecting landmarks consists of 5 phases: acquisition, preprocess, segmentation, recognition and features extraction. The output of the system is an array of features for each landmark. The first four stages are implemented using the OpenCV library. The images are grayscale and size 640x480 (Fig.).

In the preprocess phase the image (Fig. 6(a)) is transformed to facilitate the processing in the next stages. The techniques that have been used are binary thresholding (Fig. 6(b)) and morphological filtering ('closure operator') (Fig. 6(c)).

As segmentation techniques a Canny filter (Fig. 6(d)) and contour extraction (Fig. 6(e)). The next step is to extract the characteristics of each region:

- *Ratio*: number of pixels around 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 axis X, φ on Fig. 8

If a ceiling light points directly to the camera, then the acquired image will be saturated. In such cases, one big blob can be detected and the image have to be preprocessed again using a higher threshold. This situation is very frequent in the region labelled as B (Fig. 3) because lights can be very close to the camera.



(a)

(b) (c)

(d) (e)

Fig. 6. Detecting landmarks in the omnidirectional filtered image: (a) original image, (b) binary thresholded, (c) close, (d) edges and (e) contours.

IV. MAP PROJECTIONS BASED ON OMNIVISON CAMERA MODEL

In this section we present the model of our omnivision camera and how it can be used to project the objects of the environment to form an image.

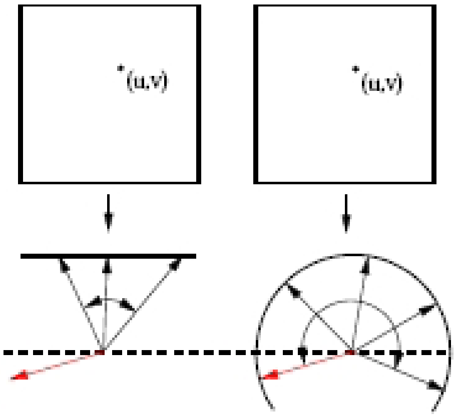


Fig. 7. The *Pin-Hole* camera model based on a flat retina (left) compared with the omnidirectional camera model based on a spherical retina (right).

A. Camera Model

The camera model describes how a 3D scene is transformed into a 2D image (Fig. 8). The standard model is the *Pin-Hole*, which projects the scene on a flat retina (Fig. 7), but it is limited to cameras with $FOV << 180^0$.

The model that best fits our system was developed by Pajdla and Bakstein [2] based on a spherical retina (Fig. 7) where the image is formed on a curved surface. In our case the radial symmetric function is:

$$r = a \cdot \tan \frac{\theta}{b} + c \cdot \sin \frac{\theta}{d}, \tag{1}$$

where *a, b, c,* and *d* are the adjustment parameters of the model, *r* is the distance in the image between the projection point of B $((u_B, v_B))$ and the image center $((u_0, v_0))$, and θ is the elevation of B with respect to the optical axis of camera (see Fig. 8).

This function makes it possible to calculate the coordinates of the image *(u,v)* depending on the azimuth (φ) and the elevation (θ) (Fig. 8):

$$\left.\begin{array}{c} u = u_0 + r \cdot cos\varphi \\ v = \beta \cdot (v_0 + r \cdot sin\varphi) \end{array}\right\} \tag{2}$$

where β is the relationship between the width and height of a pixel.

B. A Beacon Projection

If we have the coordinates of landmark i ($\mathbf{B}_i^W$) and the coordinates of the camera $\mathbf{P}$, both with respect to the environment reference system (*W*), we can calculate the projection line of the landmark $\mathbf{B}_i^P$ (B in Fig. 8) relative to the camera:

$$\mathbf{B}_i^P = Transf_{\mathbf{P}}(\mathbf{B}_i^W) = R_P * \mathbf{B}_i^W - \mathbf{P} \tag{3}$$

where R_P is the rotation matrix of $\mathbf{P}$ relative to W, i.e., the position and orientation of the camera in the environment.

To get the projection of a landmark $\mathbf{B}_i^W$, $Proj(B^P{}_i)$, in image coordinates (u_B, v_B in Fig. 8), we apply Eqs. 1 and 2:

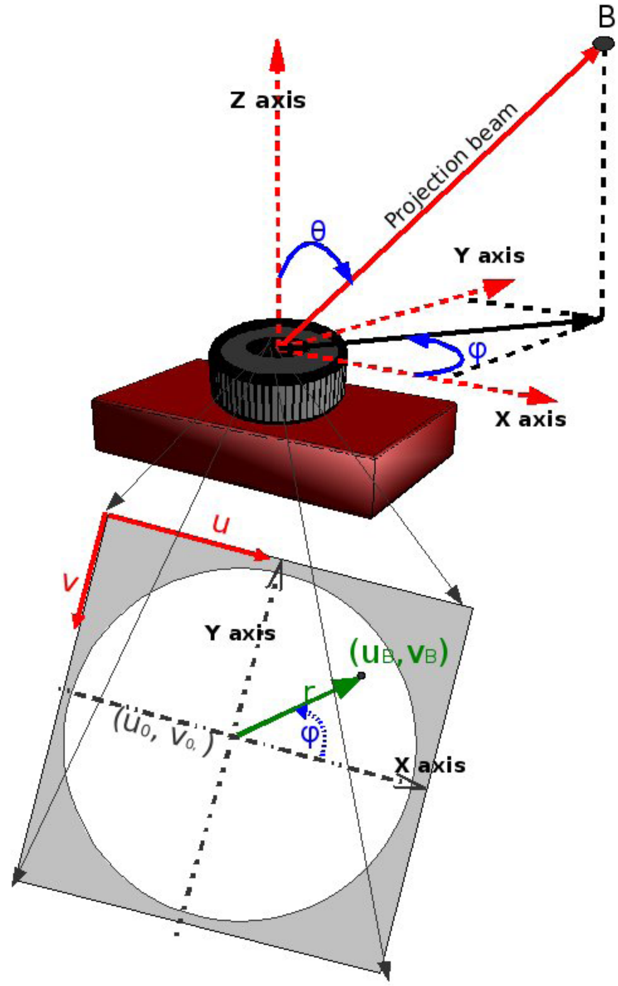


Fig. 8. Theoretic omnidirectional camera model and projection of a point B. Its projection ray is defined by the elevation (θ) and the azimuth (φ) with respect to the camera coordinate system. r y φ are the polar coordinates of the projected point (u_B, v_B). (u_0, v_0) are the coordinates of the image center.

$$Proj(B^P{}_i) = (u_{B^P{}_i}, v_{B^P{}_i}) \tag{4}$$

where Euclidean transformations were used to obtain the elevation (θ) and the azimuth (φ) angles of $\mathbf{B}_i^P$ (Fig. 8).

C. Ceiling Map Projection

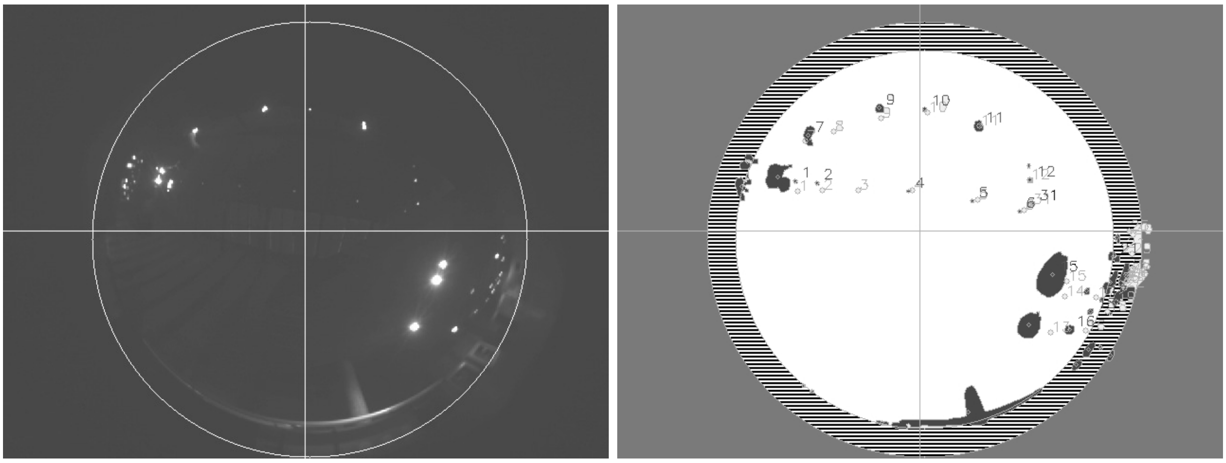
We have named ceiling map projection, $Map(P)$, to the set of theoretical positions that each mapped landmark in the environment ($B^W{}_i$) would have in the image (u, v), i.e. the pixel where it would be if the robot was at the position P.

$$Map(P) = \left\{Proj(B^P{}_i)\right\} \tag{5}$$

The algorithm is summarized in Alg. 1. A graphical example is shown in Fig. 9.

V. MONTE CARLO LOCALIZATION

Monte Carlo Localization (MCL) algorithm [15], [5] is a particle filter algorithm combined with probabilistic models of robot perception and motion. The current pose



(a) (b)

Fig. 9. Example of Ceiling Map Projection, $Map(P)$: (a) Original image; (b) Beacons projected when camera is at position **P** (gray enumeration) and landmarks detected on the image (black enumeration). Beacons in the shaded region are not considered because they are in the horizon.

Algorithm 1 Calculate $Map(P)$ for one position P.

for all Beacons ${B_i}^W$ on the map (environment reference system, W) **do**
  $\mathbf{B}_i^P = Transf_{\mathbf{P}}(\mathbf{B}_i^W)$ applying Eq.3
  $Proj(B^P{}_i) = (u_{B^P{}_i}, v_{B^P{}_i})$ applying Eqs. 1 and 2
end for

Algorithm 2 MCL Algorithm

for all m **do**
  $\overline{x}_t^m = motion\_model(u_t, x_{t-1}^m)$
  $w_t^m = measurement\_model(z_t, \overline{x}_t^m, Map)$
  $\overline{X}_t = \overline{X}_t + \langle z_t, x_t^m, Map\rangle$
end for
$X_t = resample\_model(\overline{X}_t)$
return X_t

of the robot, also called **belief**, which model a probability density function over the space of all locations. The belief about the pose space is represented with a set of discrete points in the robots's environment called particles $X_t = \{x_t^1, .., x_t^m, ..., x_t^M\}$. This type of algorithms proceeds recursively:

- A temporary particle set $\overline{X}_t$ is computed from the last particle set X_{t-1} and the last control action u_t (motion model).
- A weight factor w_t^m is assigned to each particle. It is computed based on the new sensor data at time t (last observation z_t). w_t^m is proportional to the probability that the robot is, located in x_t^m (measurement model).
- Then a new sample set X_t is calculated (resample model).

A. Motion Model

The motion model computes the probability that the robot is in state x_t, if it was previously in state x_{t-1} and control action u_t is $\overline{x}_t^m = p(x_t|u_t, x_{t-1})$.

We have applied the odometry motion model [15], where the odometry measurements are used for calculating the robot's motion over time. As we need to sample form $\overline{x}_t^m = p(x_t|u_t, x_{t-1})$, the algorithm accepts as input the previous sate, x_{t-1}, an the control $u_t = (\overline{x}_{t-1}, \overline{x}_t)$ and returns a state x_t according to $p(x_t|x_{t-1}, u_t)$ as output.

$$\begin{aligned} x_t &= (x', y', \theta') \\ x' &= x + \Delta\overline{x} + N(\overline{x}) \\ y' &= y + \Delta\overline{y} + N(\overline{y}) \\ \theta' &= \theta + \Delta\overline{theta} + N(\overline{theta}) \end{aligned} \tag{6}$$

where $\Delta\overline{x}$, $\Delta\overline{y}$,$\Delta\overline{theta}$ are the differences between the two odometry values $(\overline{x}_{t-1}, \overline{x}_t)$ and $N(\overline{x})$, $N(\overline{y})$ and $N(\overline{theta})$ are the random noise term. In this paper we modeled it with Gaussian zero-centered random variables with standard deviations $\sigma_x = 25$ cm, $\sigma_y = 25$ cm, $\sigma_\theta = 20^0$ respectively.

B. Measurement Model

The measurement model describes the probability of having a certain sensor measurement in given pose. Normally the measurements model is defined as a conditional probability distribution $p(z_t|x_t, Map)$ where x_t is the robot pose, z_t is the measurement (observation) at time t and Map is the map of the environment.

The model depends on the sensor and data used. In our case we have a omnivision camera and a map of landmarks (lights). So that we use a *Feature Based Measurement Model*. In this model z_t is the set of features extracted from the sensor measurement:

$$z_t = f(Map(P)_t) = \{f_t^1, ..., f_t^n, ..., f_t^N\} \tag{7}$$

where $f_t^n = (pu^n, pv^n)$ is the position on the image (in pixels) for each identified feature (associated landmark). The number of identified features can be different at each image (Fig. 9 and 11).

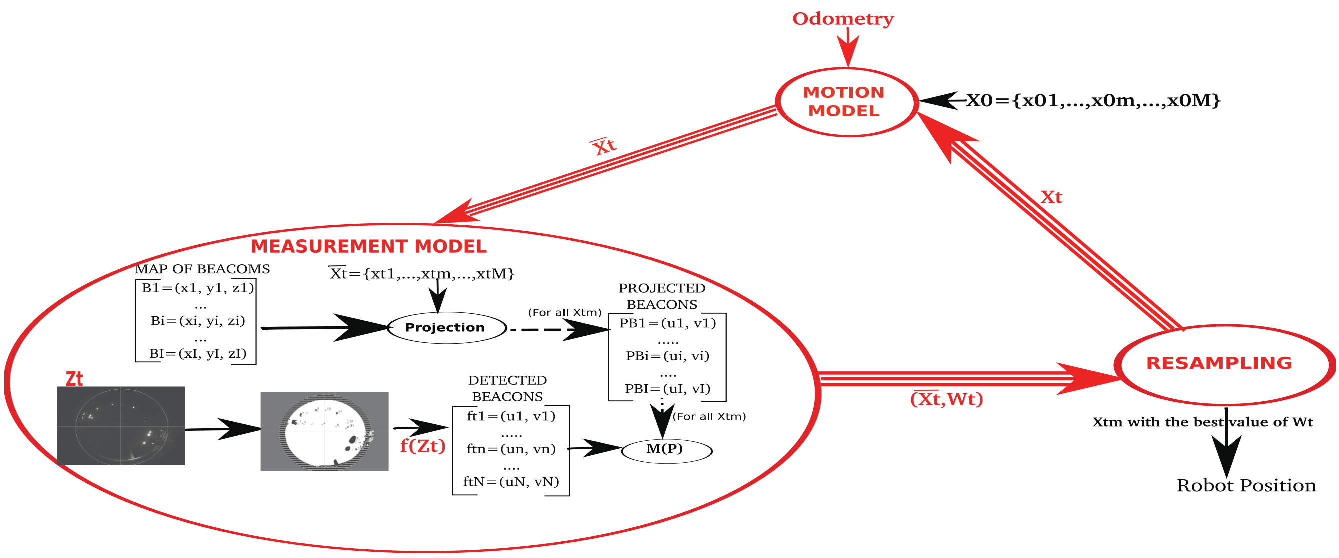


Fig. 10. General scheme of Monte Carlo localization algorithm based on omnivision and a map of landmarks. **X0** is the initial belief and **x0m** represent each initial particle (x_0^m in the text). **ftn** represent the lights detected in the image (f_t^n in the text). Each beacon in the *Map* (B_i^W) are represented as **Bi** and their projections for one particle ($Proj(B^P{}_i)$) as **PBi**.

Algorithm 3 Calculate N_P and ε_P

$Map(P)$ (Alg. 1)
for all $B^M{}_i$ in the map **do**
  $f_t^n \leftrightarrow Proj(B^P{}_i)$ (Fig. 11)
  $\varepsilon(B^P{}_i) = \|Proj(B^P{}_i) - f_t^n\|$
  if $\varepsilon(B^P{}_i) < THRESHOLD$ **then**
    $\varepsilon_P = \varepsilon_P + \varepsilon(B^P{}_i)$
    $N_P = N_P + 1$ (Number of associated landmarks)
  else
    $\varepsilon_P = \varepsilon_P + THRESHOLD$
  end if
end for

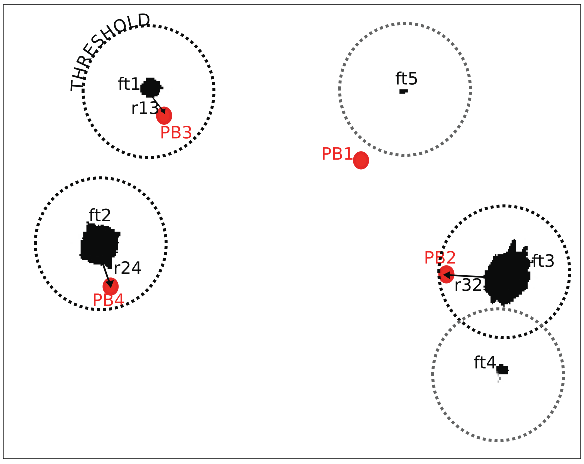


Fig. 11. Matching between detected landmarks f_t^n on the image and the ceiling map projection $Proj(B^P{}_i)$ (labeled with PBi). $N(P) = 3$ and $\varepsilon_P = r13 + r24 + r32 + THRESHOLD$.

To calculate w_t^m we need to know the *expected landmarks* ($Map(x_t^m)$) for each particle x_t^m and their likelihood with the detected landmarks (z_t)). In order to do that we define a Merit function $M(x_t^m)$ defined as (to let a clarity reading of the next equations x_t^m is denoted as P):

$$M(P) = \frac{1}{N_P} * \varepsilon_P \qquad (8)$$

where N_P is the number of identified landmarks or 'matched' landmarks and ε_P is the accumulated error (distance in pixels between detected landmarks and projected landmarks on position P, $Map(P)$ in Fig. 9). Both are calculated using Alg. 3. A graphical example of the matching process is illustrated in Fig. 11.

The best data association for each particle is the one that has the largest number of identified landmarks N_P and the smallest error ε_P in matching process

.

VI. EXPERIMENTAL VALIDATION

The experimental validation of our system has been carried out in a exposition hall at the Domus Museum located in A Coruña (Spain). It has about $24 \times 7m^2$. The experiments were performed as off-line validation tests on different sequences of images acquired in the museum. All images were labeled with the corresponding position obtained from a laser sensor. One image per second was acquired.

The results of one of this experiments can be seen on Fig. 12. The two trajectories executed by the robot are displayed on Fig. 12(a); only 2D positions (X, Y) are shown. The distance travelled in this experiment is more than 48 meters and 120 images were acquired and processed. Two consecutive images were separated at 0.40 meters on average but angular displacements can be very large. As reference, at the end of the first trajectory the robot uses only 3 steps (images from 60 to 63) to turn 180 degrees.

We use as reference the position calculated by the laser sensor when there is no people in the environment (that is an unusual situation in the museum). The maximum position error is 1.07 m but the average position error has been 0.41 m (Fig 12(b)). The maximum orientation error is 20 degrees, but only 3 degrees on average (Fig 12(c)). This angular precision is one of the main advantages of using an omnidirectional camera for localization.The irregular floor of the museum affects negatively to the position error. On the other hand, the positions of the mapped landmarks on the museum were very difficult to obtain and the errors are not negligible. Nevertheless, our system calculates correctly the position of a mobile robot on a very complex environment and over long trajectories. Its precision is enough for navigation tasks.

The time for running the algorithm can be seen in Fig. 12(d). We have used a laptop Intel Pentium 4 CPU 3.06GHz. The time for processing each image (vision time) is almost constant, except when an image is saturated. In such cases, a second landmark detection process is launched with a higher threshold (see Sect. III-A). On the other hand, localization time depends on the number of particles checked in the Monte Carlo filter. In this experiments 200 particles were checked in each step. The time required for processing each image is only 40 ms on average, so our algorithm can be executed in real time.

The proposed localization system is very robust. For example, only 10 landmarks of the 29 mapped ligths are identified on each image on average. In several steps, only 4 or 5 landmarks can be used for localization. The reasons are that many elements in the environment can occlude the lights to the camera and that lights are oriented (i.e. can not be detected on any position, see Fig. 2). In this sense, using an omnidirectional camera minimizes all these problems.

One final problem arises when there are positions in the environment from which the views of the ceiling are similar. This symmetries can mislead the system. Nevertheless, as the robot moves more information is obtained from the environment and the ambiguity can be reduced, or even eliminated, applying the Monte Carlo filter.

VII. CONCLUSIONS AND FUTURE WORK

The solution adopted in this paper to locate a guide robot on a museum is based on particle filters and a map of lights (landmarks) in the environment. The main difficulties for the localization of the robot are that the Domus museum is highly populated, the irregularity of the ground floor (swinging movement of the camera, increased odometry errors), and the height of the ceiling (measurement errors are proportional to that height).

The results of the experiments confirm the accuracy and robustness of our omnivision localization system. The position and angular errors are 0.4 m and 3 degrees on average, respectively. In spite of the irregular floor of the museum and the imprecision on the mapped landmarks.

The proposed algorithm can be executed very efficiently. On one hand, using a IR filter simplifies the process of landmark detection on the images. On the other hand, the Monte Carlo filter reduces the computations needed to integrate observations and control information over time.

We want to highlight that our system was designed to cope with occlusions. The very wide visual field guarantees that enough landmarks will be viewed to estimate a good localization. In our experiments only 31 lights are considered and the robot can be located in a space of 168 m^2.

Currently our studies are focussed on the implementation of an omnivision SLAM.

VIII. ACKNOWLEDGMENTS

Authors wish to especially thank the collaboration of Tomas Pajdla and Hynek Bakstein in the calibration of the camera. This work has been financed by the Galician regional government and the Spanish Government through grants PGIDIT-TIC206011PR, TIN2005-03844.

REFERENCES

[1] ANDREASSON, H., TREPTOW, A., AND DUCKETT, T. Localization for mobile robots using panoramic vision, local features and particle filter. pp. 3348–3353.

[2] BAKSTEIN, H., AND PAJDLA, T. Panoramic mosaicing with a 180° field of view lens. In *OMNIVIS '02: Proceedings of the Third Workshop on Omnidirectional Vision* (2002), pp. 60–67.

[3] BRASSART, E., DELAHOCHE, L., CAUCHOIS, C., DROCOURT, C., PEGARD, C., AND MOUADDIB, E. M. Experimental results got with the omnidirectional vision sensor: Syclop. *Omnivis* (2000), 145.

[4] CAO, Z. L., OH, S. J., AND HALL, E. L. Dynamic omnidirectional vision for mobile robots. In *Journal of Robotic Systems* (1986), pp. 5–17.

[5] CHOSET, H., LYNCH, KEVIN, M., HUTCHINSON, S., KANTOR, G., BURGARD, W., KAVRAKI, L. E., AND THRUN, S. *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT Press, 2005.

[6] CIVERA, J., DAVISON, A., AND MONTIEL, J. Inverse depth to depth conversion for monocular slam. *Robotics and Automation, 2007 IEEE International Conference on* (2007), 2778–2783.

[7] DAVISON, A. Real-time simultaneous localisation and mapping with a single camera. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on 2* (2003), 1403–1410.

[8] LEMAIRE, T., AND LACROIX, S. Slam with panoramic vision. *Journal of Field Robotics 24*, 1-2 (2007), 91–111.

[9] MENEGATTI, E., PRETTO, A., SCARPA, A., AND PAGELLO, E. Omnidirectional vision scan matching for robot localization in dynamic environments. In *IEEE transactions on robotics* (2006), vol. 22.

[10] MURILLO, A. C., SAGÜÉS, C., GUERRERO, J. J., GOEDEMÉ, T., TUYTELAARS, T., AND GOOL, L. V. Hierarchical localization by matching vertical lines in omnidirectional images. In *From Sensors to Human Spatial Concepts workshop, held together with IROS 06* (2006), pp. 13–19.

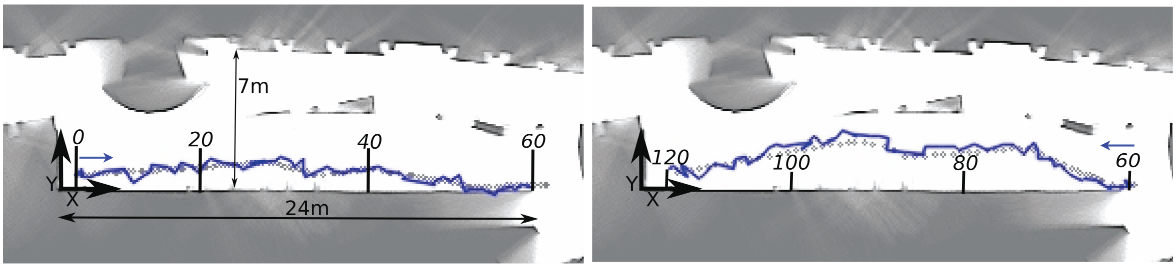
[11] PAZ, L. M., PINIS, P., TARDS, J., AND J., N. 6dof slam with stereo-in-hand. In *Visual SLAM Workshop* (2007).

[12] RODA, J. P., SAEZ, J. M., AND ESCOLANO, F. Ceiling mosaics through information-based slam. *Intelligent Robots and Systems, IEEE/RSJ International Conference on* (2007), 3898–3904.

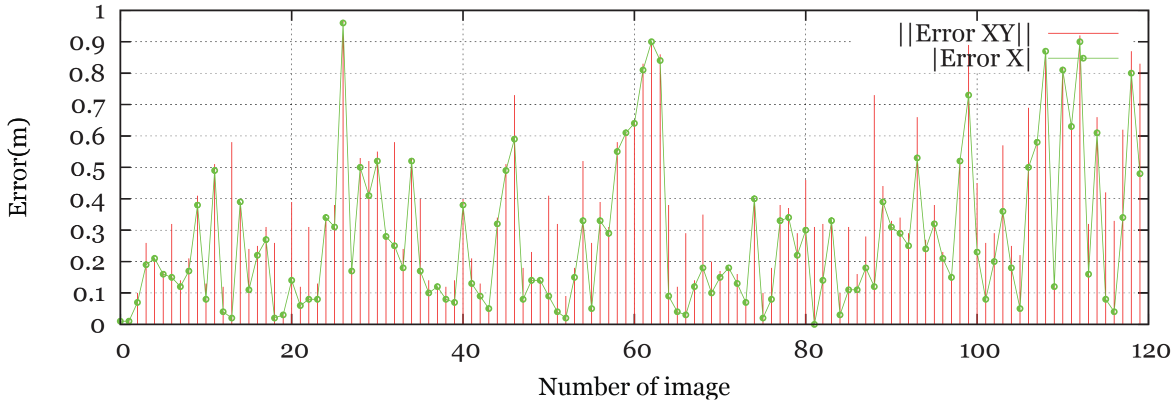
[13] SUN, Y., CAO, Q., AND CHEN, W. An object tracking and global localization method using omnidirectional vision system. In *5th World Congress on Intelligent Control and Automation* (Hangzhou. P.R. China, 2004).

[14] THRUN, S., BEETZ, M., BENNEWITZ, M., CREMERS, A., DELLAERT, F., FOX, D., HÄHNEL, D., ROSENBERG, C., ROY, N., SCHULTE, J., AND SCHULZ, D. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research* (2000).

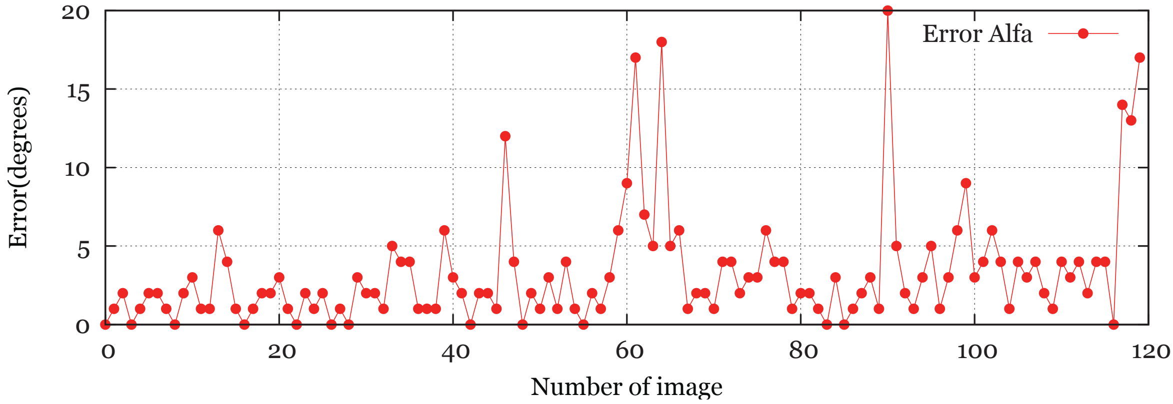
[15] THRUN, S., BURGARD, W., AND FOX, D. *Probabilistic Robotics.* MIT Press, 2003.



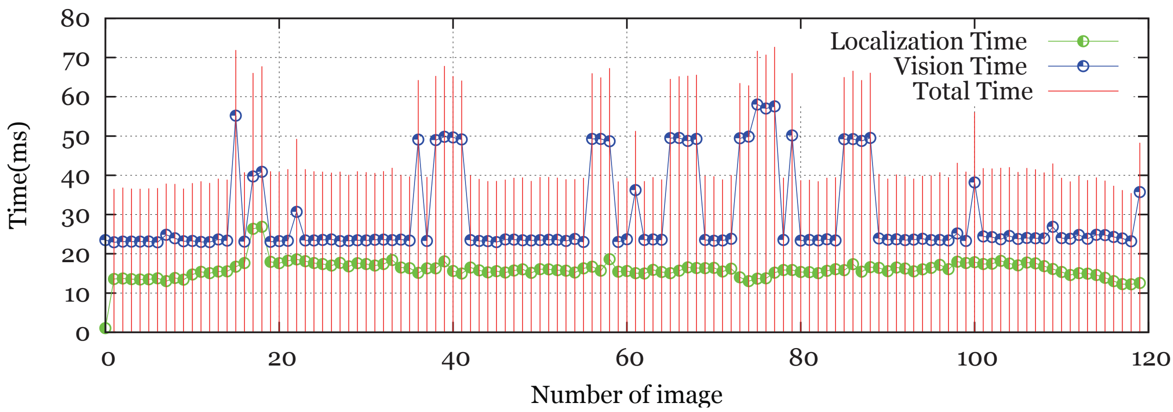
(a)



(b)



(c)



(d)

Fig. 12. Experiments on the Domus Museum: (a) Omnivision localization (gray) plotted on the grid map created with laser data (real trajectory is marked with dark circles). Position (b) and orientation error (c) between laser pose estimation and the omnivision localization. (d) Time for processing each image.