An Evaluation of Weighting Methods for Appearance-based Monte Carlo Localization using Omnidirectional Images

Lorenzo Fernandez, Arturo Gil, Luis Paya, Oscar Reinoso

***Abstract*— In this paper we deal with the problem of mobile robot localization using omnidirectional images. We assume that the robot is equipped with an omnidirectional camera. In addition, we consider that the map consists of a set of omnidirectional images with known positions in the environment. Each omnidirectional image is represented by a single Fourier descriptor that represents the appearance as well as the orientation. Given an image captured with the camera at a certain time, the Fourier descriptor allows us to find the image in the map that is most similar in appearance. We propose the use of Monte Carlo localization to estimate the most probable pose of the robot. Based on these assumptions, in this paper we propose several methods that allow to compute a weight for each particle and carry out a comparison in terms of the error in localization. Experimental results are presented using indoor omnidirectional images and a real robotic platform.**

I. INTRODUCTION

An essential problem in mobile robotics considers the computation of the situation of the vehicle in a given environment. Knowing its location in the space is crucial for an autonomous agent, since the pose is needed for a precise navigation. During the last decades the Monte Carlo algorithm has been extensively used in localization tasks in the field of mobile robotics, demonstrating a large degree of robustness and efficiency ([2], [4], [11]). The approaches to Monte Carlo localization differ basically in the nature of the sensor installed on the robot. For example in [11] a laser range sensor is used to localize the robot in the environment. The differences between the expected and the current laser measurements are used to weight the particles and discard the most unlikely ones. In [2], a camera pointing to the ceiling of the environment was used. The Monte Carlo algorithm tested the brightness in the center of the images with an image of the ceiling as seen from below. The difference between the expected and actual illumination was used to weight the particles and estimate the position of the robot. In [4] a stereo camera system is used to obtain observations from a set of point features in the environment. The observed distance computed from the stereo images is used to refine the robot location in the map.

In this paper we deal with the problem of mobile robot localization using omnidirectional images. During the last years, installing omnidirectional cameras on robots has become common, due to its low cost, weight and power consumption and due to the fact that cameras are able to provide a high quantity of information. In this paper, we consider that the camera is installed at a fixed orientation with respect to the robot and pointing upwards in direction to an omnidirectional mirror. We also assume that the movement of the robot is restricted to a plane. In this case, a rotation of the robot corresponds to a shift in the columns of the panoramic image.

Several authors have investigated the use of omnidirectional images for robot localization. These solutions can be divided into two main groups:

- Feature-based solutions, in which a number of significant points from each omnidirectional image are extracted. Next, each point in the image is described using an invariant descriptor. For example [8] uses omnidirectional images to find the location of the robot in a given map, using SURF features [1].
- Appearance-based solutions, in which the whole appearance of the image is represented by a single descriptor ([6], [5]).

In the latter case, the methods used to obtain the localization of the robot based on an observation function are diverse. In addition, the descriptors used in each research are different, thus making difficult the comparison of the methods. This fact justifies the analysis presented here. In consequence, in this paper we present a study of different methods that allow to compute a weight for each particle and carry out a comparison in terms of the error in localization. Some of the methods presented here have been used in the past for localization tasks using particle filters (e.g. [7]), but, in addition, we present some techniques that in some cases are able to provide also good results in terms of accuracy. The results presented also evaluate the influence of the descriptor in the localization.

The work presented here differs mainly from prior works in the two following aspects: First, the map is represented by a grid, where several omnidirectional images are taken at certain positions of the environment. Other authors represent the environment taking images during a given trajectory of the robot [7]. Second, we propose different methods that allow to localize the robot using a particle filter and new methods that allow to refine the position of the robot rapidly.

We describe each omnidirectional image by a single Fourier descriptor that represents the appearance with invariance to the rotation. We have chosen this descriptor based on a prior work [9], in which the Fourier descriptor allows a fast comparison between the current image and the map by means of a vector distance measurement. Finally, in general

This work has been supported by the Ministerio de Educacion y Ciencia, by means of project number DPI2007-61197.

Lorenzo Fernandez, Arturo Gil, Luis Paya and Oscar Reinoso are with Universidad Miguel Hernndez de Elche, Systems Engineering Department, Avda. de la Universidad s/n, Elche (Alicante), SPAIN {l.fernandez|arturo.gil|lpaya|o.reinoso}@umh.es

Proceedings of the 2nd. Workshop on Omnidirectional Robot Vision
A workshop of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)
Anchorage, Alaska, USA, May 7, 2010, ISBN 978-88-95872-02-5
pp. 13-18

the processing time needed to compute the Fourier transform is comparable to common feature extraction and description methods.

The methods described here are in fact independent of the descriptor used to represent the images. In particular, other correspondence methods such us those based on image features [8] may also be applied, or other appearance-based methods such as PCA [5].

The rest of the paper is organized as follows: section II describes the Fourier transform and its use with omnidirectional images. Section III deals with the Monte-Carlo algorithm and its application to the problem of localization in mobile robotics. Next, Section IV lists the weight methods studied to represent the likelihood of observations during Monte Carlo localization. Section V presents the results obtained. Finally, Section VI exposes the main conclusions and proposes future work.

II. FOURIER SIGNATURE WITH OMNIDIRECTIONAL IMAGES

To date, different description methods have been used in the context of omnidirectional robot vision. In this work, we make use of Fourier-based techniques. When we have an image $f(x, y)$ with N_x rows and N_y columns, we can obtain the most relevant information from the image by means of the Discrete Fourier Transform.

There are several possibilities, such as to implement the 2D Discrete Fourier Transform [9], the Spherical Fourier Transform of omnidirectional images [10] or the Fourier Signature of the panoramic image [6]. The Fourier signature exploits better the invariance to ground-plane rotations in panoramic images. This transformation consists in expanding each row of the panoramic image $\{a_n\} = \{a_0, a_1, \ldots, a_{N_y-1}\}$ using the Discrete Fourier Transform into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \ldots, A_{N_y-1}\}$. The most important information is concentrated in the low frequency components of each row, so we can work only with the information from the k first columns in the Signature. Also, this feature presents rotational invariance. It is possible to prove that if each row of the original image is represented by the sequence $\{a_n\}$ and each row of the rotated image by $\{a_{n-q}\}$ (being q the amount of shift), when the Fourier Transform of the shifted sequence is computed, we obtain the same amplitudes A_k than in the non-shifted sequence, and there is only a phase change, proportional to the amount of shift q, (eq.1).

$$F[\{a_{n-q}\}] = A_k \exp -j\frac{2\pi qk}{N_y}; \quad k = 0, \ldots, N_y - 1 \quad (1)$$

Thanks to this shift Theorem, we can separate the computation of the robot position and the orientation. It is interesting to highlight also that the Fourier Signature is an inherently incremental method (what differs from the PCA Analysis).

III. MONTE-CARLO LOCALIZATION

In robot localization we are interested in the estimation of the pose of the vehicle (typically, the state $x_t = (\mathrm{x},\ \mathrm{y},\ \theta)$) at time t using a set of measurements $z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ from the environment and the movements $u_{1:t} = \{u_1, u_2, \ldots, u_t\}$ of the robot. In this notation, we consider that the robot makes a movement u_t from time $t-1$ to time t and next obtains an observation z_t. The localization problem can be stated in a probabilistic way: we aim at estimating a probability function $p(x_t|z_{1:t}, u_{1:t})$ over the space of all possible poses, conditioned on all data available until time t, the observations $z_{1:t}$, movements performed $u_{1:t}$ and the map. The estimation process is usually carried out in two phases:

Prediction phase: The motion model is used to compute the probability function $p(x_t|z_{1:t-1}, u_{1:t})$, taking only motion into account. Generally, we assume that the current state x_t depends only on the previous state x_{t-1} and the movement u_t. The motion model is specified in the form of the conditional density $p(x_t|x_{t-1}, u_t)$. The probability function at the next step is obtained by integration:

$$p(x_t|z_{1:t-1}, u_t) = \int p(x_t|x_{t-1}, u_t).p(x_{t-1}|z_{1:t-1}, u_{1:t-1})dx_{t-1}, \quad (2)$$

where the function $p(x_t|x_{t-1}, u_t)$ represents the probabilistic movement model.

Update phase In the second phase, a measurement model is used to incorporate information from the sensors and obtain the posterior distribution $p(x_t|z_{1:t}, u_{1:t})$. In this step, the measurement model $p(z_t|x_t)$ is employed, which provides the likelihood of obtaining the observation z_t assuming that the robot is at pose x_t. The posterior $p(x_t|z_{1:t}, u_{1:t})$, can be calculated using Bayes' Theorem:

$$p(x_t|z_{1:t}, u_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1}, u_t)}{p(z_t|z_{1:t-1})} \quad (3)$$

This process is repeated recursively after the update phase. The knowledge about the initial state at time t_0 is generally represented by $p(x_0)$. In this case two different cases are generally considered:

- The case of global localization, in which the initial pose of the vehicle is totally unknown. Thus $p(x_0)$ is represented by a uniform distribution.
- The case of local localization or tracking, in which the initial pose of the vehicle is partially known. The function $p(x_0)$ is commonly represented by a gaussian distribution centered at the known starting pose of the robot.

Note that in Equations (2) and (3) nothing is said about the representation of the probability function. We concentrate on the Monte Carlo localization method, that can be included in a set of algorithms called particle filters, extensively used during last decade (e.g. [3], [11]). In Monte Carlo Localization (MCL), the probability density function $p(x_t|z_{1:t}, u_{1:t})$ is represented by a set of M random samples $\chi_t = \{x_t^i, i = 1 \ldots M\}$ extracted from it, named particles. Each particle can be understood as a hypothesis of the true state of the robot $x_t^i = (\mathrm{x}^i,\ \mathrm{y}^i,\ \theta^i)$. The algorithm is described in the next lines, and consists of a prediction and update phase:

Proceedings of the 2nd. Workshop on Omnidirectional Robot Vision
A workshop of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)
Anchorage, Alaska, USA, May 7, 2010, ISBN 978-88-95872-02-5
pp. 13-18

Prediction Phase: At time t a set of particles $\overline{\chi_t}$ is generated based on the set of particles χ_{t-1} and a control signal u_t. This step uses the motion model $p(x_t|x_{t-1}, u_t)$. In order to represent this probability function, the movement u_t is applied to each particle while adding a pre-defined quantity of noise. As a result, the new set of particles $\overline{\chi_t}$ represents the density $p(x_t|z_{1:t-1}, u_{1:t})$.

Update Phase: In this second phase, the observation z_t obtained by the robot is used to compute a weight ω_t^i for each particle in the set $\overline{\chi_t}$. This weight represents the observation model $p(z_t|x_t)$ and is computed as $\omega_t^i = p(z_t|x_t^i)$. In this paper we propose different methods for the computation of this weight, that will be described in Section IV. The weights are normalized so that $\sum \omega_t^i = 1$. As a result, we obtain a set of particles accompanied by a weigth $\overline{\chi}_t = \{x_t^i, \omega_t^i\}$.

Finally the resulting set χ_t is calculated by resampling with replacement from the set $\overline{\chi}_t$, where the probability of resampling each particle is proportional to its importance weight ω_t^i. Finally, the set χ_t represents the distribution $p(x_t|z_{1:t}, u_{1:t})$.

IV. WEIGHT METHODS

As described in the previous section, the Monte-carlo Algorithm introduces the current observation z_t of the robot by means of computing a weight w_i for each particle and performing a resampling process.

We consider that our map is formed by a set of N bi-dimensional landmarks $L = \{l_1, l_2, \ldots, l_N\}$, forming a grid in the environment with a particular resolution. Each landmark lj has an omnidirectional image I_j associated and a Fourier descriptor d_j that describes the global appearance of the image, thus $l_j = \{(l_{j,\mathrm{x}}, l_{j,\mathrm{y}}), d_j, I_j\}$.

Next, we describe the localization method proposed. We consider that at time t the robot captured an image and computed the Fourier descriptor d_t. Using this Fourier descriptor we compare the descriptor d_t with the rest of descriptors $d_j,\ j = 1 \ldots N$ and find the B landmarks in the map that are closest in appearance with the current image I_t. In this sense, we allow the correspondence of the current observation to several landmarks in the map. We consider that this is a special case of the data association problem. In addition, this correspondence benefits the localization algorithm, since it may restrict the computation of the observation model to a reduced set of landmarks, thus reducing the computational effort. We will show results when varying this parameter in order to assess its influence. In addition the selection of B landmarks in terms of appearance will allow us to evaluate the importance of the description method used.

We base the localization of the robot on the Monte Carlo algorithm explained in the previous section. Next, we propose several methods that allow to compute the weight of each particle $\omega_t^i = p(z_t|x_t^i)$, thus providing different observation models:

- Weight Method 1 (W1): product of gaussians centered on each image landmark considering the distance to the descriptor.

$$\omega_t^i = \prod_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \tag{4}$$

where, $v_j = (l_{j,\mathrm{x}}, l_{j,\mathrm{y}}) - (\mathrm{x}^i, \mathrm{y}^i)$ is the difference between the position of the landmark l_j and the position $(\mathrm{x}^i, \mathrm{y}^i)$ of the particle i. The matrix Σ_l is a diagonal matrix $\Sigma_l = diag(\sigma_l^2, \sigma_l^2)$. The variance σ_l^2 is chosen experimentally in order to minimize the error in the localization. We recall that the product of gaussian distributions is also a gaussian. The results demonstrate that this method tends to center the particles rapidly near the true robot pose, however, it suffers from some problems when the data association phase fails (e.g. the selected landmark l_j lies far away from the actual robot pose).
- Weight Method 2 (W2): Sum of gaussians centered on each image landmark.

$$\omega_t^i = \sum_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \tag{5}$$

where v_j and matrix Σ_l is analogous to the matrix defined in W1 and its values were also selected experimentally. In this case, the observation model $p(z_t|x_t)$ is not gaussian, since it is formed by a sum of gaussians, being thus multi-modal. As we will show in the experimental results, this method is less sensitive to errors in data association whereas it is able to achieve nice localization results.
- Weight method 3 (W3): sum of gaussians centered on each image landmark and considering the difference in the descriptors.

$$\omega_t^i = \sum_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \exp\{-h_j \Sigma_d^{-1} h_j^T\} \tag{6}$$

where v_j and Σ_l have been defined in the previous methods and $h_j = |d_j - d_t|$ defines the difference between the module of the Fourier descriptor associated to the current image observed and the module of the descriptor associated to the landmark l_j. The descriptors are normalized so that the summation of the euclidean distance of the current descriptor d_t to the rest of the B associations equals one, $\sum_{j=1}^{B} h_j = 1$. The matrix $\Sigma_d = diag(\sigma_d^2)$ is an $k \times k$ matrix, being k the length of the Fourier descriptor. The main difference of this method with respect to W2 is the consideration of the difference in the descriptor in the observation model $p(z_t|x_t)$. This fact generally gives higher weights to particles situated near a landmark that is close in appearance to the current observation.
- Weight method 4 (W4): product of gaussians centered on each image landmark and considering the difference in the descriptors.

$$\omega_t^i = \prod_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \exp\{-h_j \Sigma_d^{-1} h_j^T\} \tag{7}$$

Proceedings of the 2nd. Workshop on Omnidirectional Robot Vision
A workshop of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)
Anchorage, Alaska, USA, May 7, 2010, ISBN 978-88-95872-02-5
pp. 13-18

where v_j, h_j, Σ_l and Σ_d have been defined in the previous methods. This method is similar to method 1 but considering the effect of the similarity in the description when computing the weight.

- Weight method 5 (W5): sum of gaussians centered on each landmark position and considering the difference in the descriptors as well as the orientation of the landmarks (images).

$$\omega_t^i = \sum_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \cdot \cdot \exp\{-h_j \Sigma_d^{-1} h_j^T\} \cdot \exp\{-g_j \Sigma_\theta^{-1} g_j^T\} \quad (8)$$

where v_j, h_j, Σ_l and Σ_d have been defined in the previous methods. The variable $g_j = (\theta_j - \theta_i)$ computes the difference between the expected orientation θ_j and θ_i the orientation of the particle. Given the current descriptor d_t and the descriptor d_j the orientation θ_j can be computed as in Equation (1). In this case, and since the map is known, the orientation of all the landmarks (images) in the map is known in advance. The matrix Σ_θ is selected experimentally.

- Weight method 6 (W6): product of gaussians centered on each landmark position and considering the difference in the descriptors as well as the orientation of the landmarks (images).

$$\omega_t^i = \prod_{j=1}^{B} \exp\{-v_j \Sigma_l^{-1} v_j^T\} \cdot \cdot \exp\{-h_j \Sigma_d^{-1} h_j^T\} \cdot \exp\{-g_j \Sigma_\theta^{-1} g_j^T\} \quad (9)$$

where v_j, h_j, g_j, Σ_l, Σ_d and Σ_θ have been defined in the previous methods. This method is similar to the W5, but considering the product of the gaussian distributions.

- Weight method 7 (W7): gaussian distribution at the center of mass of a discrete particle system. This method is inspired in a system of particles, each one having a mass related to the similarity with the current descriptor d_t observed by the robot. The weight for each particle is computed as:

$$\omega_t^i = \exp\{-f_j \Sigma_f^{-1} f_j^T\} \quad (10)$$

where $f_j = ((\mathrm{x}^i, \mathrm{y}^i) - \hat{c})$ computes the difference between the position of the particle i and the center of mass computed as:

$$\hat{c} = \sum_{j=1}^{B} l_j \cdot m_j \quad (11)$$

where the virtual mass m_j is computed as $m_j = \exp\{-h_j \Sigma_d^{-1} h_j^T\}$. The masses m_j are normalized so that $\sum_{j=1}^{B} m_j = 1$. The covariance matrix Σ_f is computed as the covariance associated to $\hat{c}$.

- Weight method 8 (W8): gaussian distribution at the center of a spring-mass system. This method is inspired by a spring-mass system [6]. The constant of each spring is related to the similarity in the description, thus, landmarks more similar to the current observation try to atract the mass more tightly. To simplify the calculations, m_j is equal to 1 for all the mass of the system. The weight for each particle is computed as:

$$\omega_t^i = \exp\{-f_j \Sigma_s f_j^T\} \quad (12)$$

where $f_j = ((\mathrm{x}^i, \mathrm{y}^i) - \overline{c})$ computes the difference between the position of the particle i and the center $\overline{c}$ of a spring-mass system. In this case, the matrix Σ_s is computed as the covariance associated to $\overline{c}$.

- Weight method 9 (W9): triangular distribution. This method is inspired in the weight function introduced by [7]. The weight for each particle is computed as:

$$\omega_t^i = \frac{1}{B} \sum_{j=1}^{B} S_j (D_{max}^i - \sqrt{v_j v_j^T}) \quad (13)$$

where $S_j = (1 - \sqrt{h_j h_j^T})$ and $v_j = (l_{j,\mathrm{x}}, l_{j,\mathrm{y}}) - (\mathrm{x}^i, \mathrm{y}^i)$ computes the difference between the position of the particle i and the landmark j. D_{max}^i is the metric distance between the farthest landmark and the position of the particle i. This weight method represents a triangular distribution centered on each landmark as to the appearance of each acquired image.

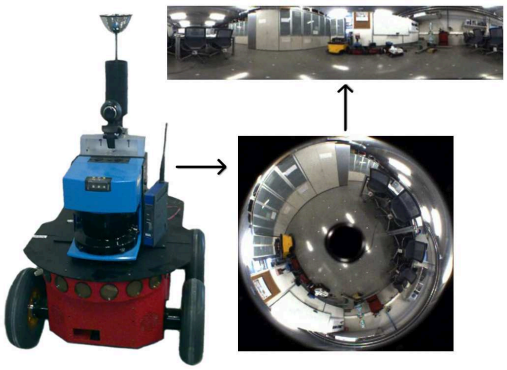


Fig. 1. Pioneer P3-AT mobile robot, omnidirectional image and panoramic image .

V. EXPERIMENTS

In order to acquire data for our experiments we have used a Pioneer P3-AT mobile robot, equipped with an omnidirectional camera (Figure 1). The map was built by carefully obtaining omnidirectional images at different positions of an office-like environment. Next, the robot performed a trajectory inside the map, gathering omnidirectional images whenever it traversed a distance above $0,1m$. In order to obtain a robust result, we have performed different types of trajectories, varying both the number of images captured and the angles between them. The map is formed by a set of images placed in a grid with a resolution of $0,2m$. The position of this set of images of the map is represented with black circles in Figure 2(a)(b)(d)(e).

Proceedings of the 2nd. Workshop on Omnidirectional Robot Vision
A workshop of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)
Anchorage, Alaska, USA, May 7, 2010, ISBN 978-88-95872-02-5
pp. 13-18

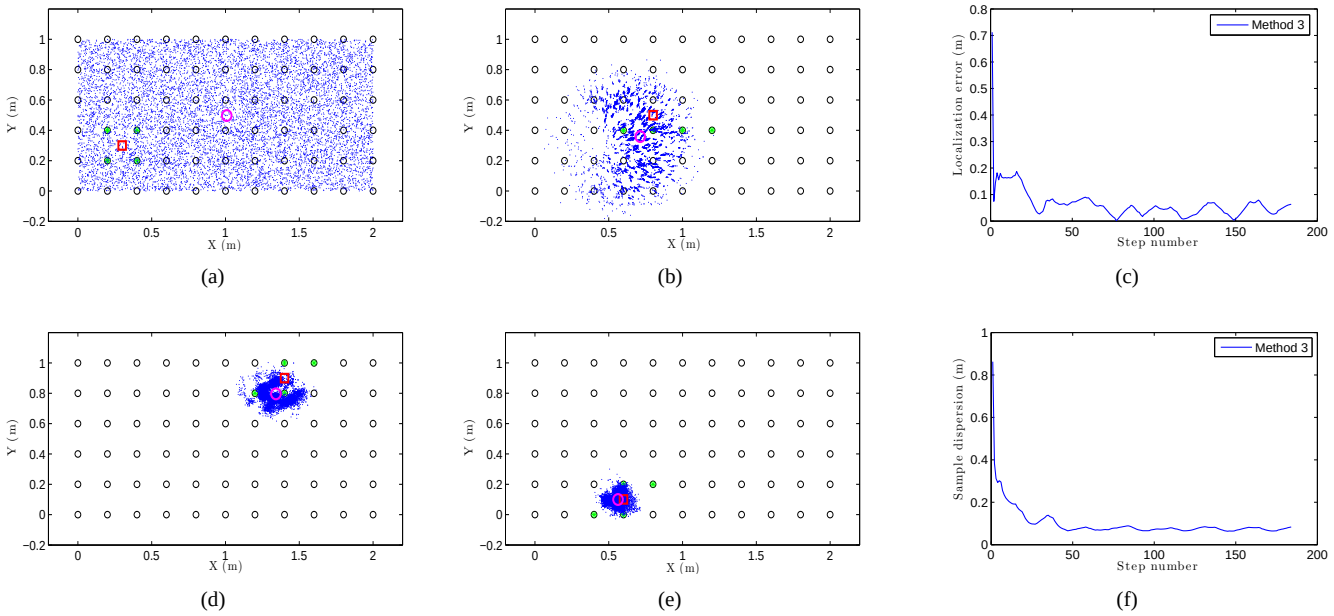


Fig. 2. (a) and (b) An example of Monte-Carlo localization using the W3 and 10000 particles performing a global localization and (d) and (e) then position tracking, (c) error in location for the previous experiment and (f) error in orientation for the previous experiment.

Figure 2 presents an example of global localization using the method W3 using 10000 particles. The position of the landmarks is indicated with a green circle. Figure 2(a) presents an uniform distribution of the particles on the environment. In the sequence presented in Figure 2(a)(b)(d)(e) we can observe how the particles concentrate near the true pose of the robot. Please note that the trajectory of the robot does not coincide exactly with the position of any of the images in the map. In consequence, the localization can be performed without having to place the robot exactly on a landmark. Figure 2(c) presents the error in position at each iteration step, the figure shows that, as the robot moves around the map, the error in location is reduced and so does the dispersion of particles, as can be observed in Figure 2(f).

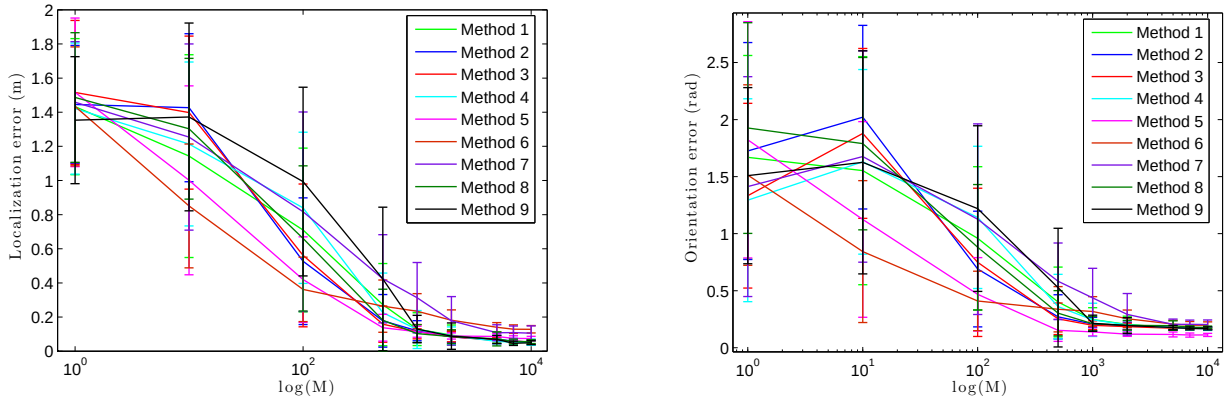
To compare different types of weighting, we have carried out a series of experiments of global localization in which we have obtained the trajectory average error in position and orientation of the robot depending on the number of particles M. As shown in Figure 3 as we increase the number of particles, the error decreases both in location and orientation. We have observed that the method W5 is able to achieve nice localization results even with a low number of particles.

To compare the weighting methods with respect to the number of associations, we have separated the global location from the tracking. Figure 4a) presents the error in tracking when varying the number of associations B. As shown in Figure 4a), in general, although the number of associations is low, the error in localization remains small. As the number of associations increases, the error in the location is small in the sum-of-gaussian methods, but increases rapidly in the product of gaussian methods (W2, W4 and W6). When we multiply two gaussians we get a gaussian with variance lower than the minimum variance of both. Therefore, as the number of associations grows, the weighting of particles becomes more restrictive. On the other hand, Figure 4b) presents the results in global localization. When the number of associations is increased, the error in the location grows quicker comparing to the case of tracking (Figure 4a)). Moreover, we observe that the last 3 methods (W7, W8 and W9) require a minimum number of associations to work properly. Finally, we can observe how when we work with the method W1, the error also increases with increasing the number of associations. This occurs because the method W1 does not take into account the descriptor, using only the data association.

VI. CONCLUSION

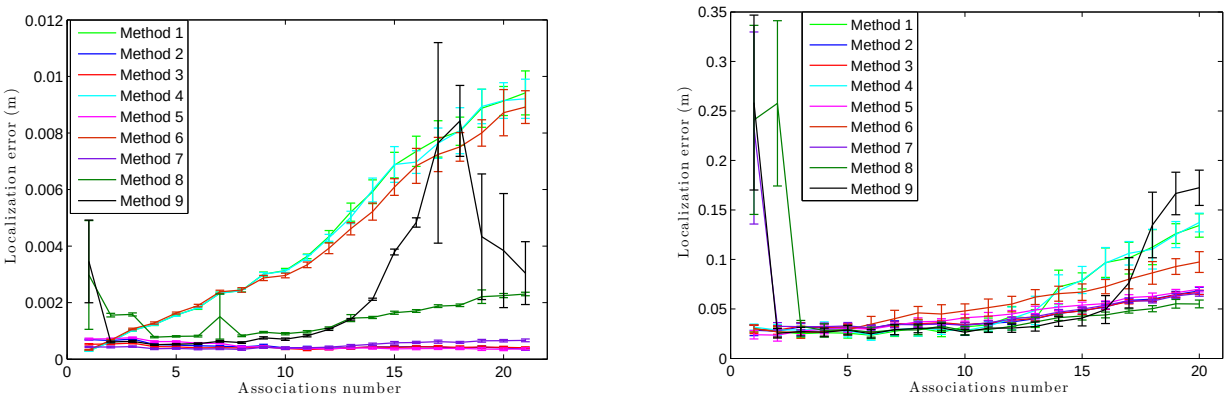
In this work, we have exposed an appearance-based Monte Carlo localization method using omnidirectional images and we have compared different weighting methods. We have built the image descriptors using the Fourier signature of panoramic images. We have evaluated the performance of the different weight methods in the case of local and global localization finding different behaviours. We have also evaluated the influence of the descriptor in the localization by varying the number of possible associations. Our system is able to track the position of the robot while moving and it is able to estimate the position of the robot in the case of unknown initial position. We proved how the precision of the methods varies with the type of weight, the number of particles used and the number of associations. In the evaluated methods, as we increase the number of particles in the system, the average error of localization decreases rapidly. With respect to the orientation, we obtained a similar results. We have demonstrated the strong dependence between the number of associations and the type of method used. We prove

Proceedings of the 2nd. Workshop on Omnidirectional Robot Vision
A workshop of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)
Anchorage, Alaska, USA, May 7, 2010, ISBN 978-88-95872-02-5
pp. 13-18



(a) (b)

Fig. 3. (a) Trajectory average error in position and (b) orientation versus the number of particles M of different methods of weighted.



(a) (b)

Fig. 4. Trajectory average error in position versus associations number for all methods using 10.000 particles. (a) Tracking and (b) global localization

how it is possible to correct the weighting of the particles by combining a physical system of forces with a Gaussian weight (W8).

This work opens the door to new applications of the appearance-based methods in mobile robotics. Once we have shown how it is possible to perform an appearance-based Monte Carlo localization with different types of weighting, we are now working on new SLAM applications using appearance-based methods.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, 2006.

[2] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. *IEEE Computer Society Conference on Computer VIsion and Pattern Recognition (CVPR99)*, 1999.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. *ICRA*, 1999.

[4] A. Gil, O. Reinoso, M. A. Vicente, C. Fernández, and L. Payá. Monte carlo localization using SIFT features. *Lecture Notes in Computer Science*, I(3523):623–630, 2005.

[5] M. Jogan and A. Leonardis. Robust localization using eigenspace of spinning-images. In *Proc. of the IEEE Workshop on Omnidirectional Vision*, pages 37–44, Hilton Head Island, USA, 2000.

[6] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–276, 2004.

[7] E. Menegatti, M. Zocaratto, E. Pagello, and H. Ishiguro. Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[8] A. C. Murillo, J. J. Guerrero, and C. Sagüés. Surf features for efficient robot localization with omnidirectional images. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, San Diego, CA, USA, 2007.

[9] L. Paya, L. Fernandez, O. Reinoso, A. Gil, and D. Ubeda. Appearance-based dense maps creation. comparison of compression techniques with panoramic images. In *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics.*, pages 238–246, Milan, Italy, 2009.

[10] F. Rossi, A. Ranganathan, F. Dellaert, and E. Menegatti. Toward topological localization with spherical fourier transform and uncalibrated camera. In *Proc. of the Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots.*, pages 319–330, Venice, Italy, 2008.

[11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.