Sensors **2015**, *15*, 26368-26395; doi:10.3390/s151026368

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

Position Estimation and Local Mapping Using Omnidirectional Images and Global Appearance Descriptors

Yerai Berenguer *, Luis Payá, Mónica Ballesta and Oscar Reinoso

Departamento de Ingeniería de Sistemas y Automática, Miguel Hernández University,
Avda. de la Universidad s/n, Elche (Alicante) 03202, Spain; E-Mails: lpaya@umh.es (L.P.);
m.ballesta@umh.es (M.B.); o.reinoso@umh.es (O.R.)

* Author to whom correspondence should be addressed; E-Mail: yberenguer@umh.es;
Tel.: +34-96-522-2435; Fax: +34-96-665-8979.

Academic Editor: Kourosh Khoshelham

Received: 27 July 2015 / Accepted: 14 October 2015 / Published: 16 October 2015

Abstract: This work presents some methods to create local maps and to estimate the position of a mobile robot, using the global appearance of omnidirectional images. We use a robot that carries an omnidirectional vision system on it. Every omnidirectional image acquired by the robot is described only with one global appearance descriptor, based on the Radon transform. In the work presented in this paper, two different possibilities have been considered. In the first one, we assume the existence of a map previously built composed of omnidirectional images that have been captured from previously-known positions. The purpose in this case consists of estimating the nearest position of the map to the current position of the robot, making use of the visual information acquired by the robot from its current (unknown) position. In the second one, we assume that we have a model of the environment composed of omnidirectional images, but with no information about the location of where the images were acquired. The purpose in this case consists of building a local map and estimating the position of the robot within this map. Both methods are tested with different databases (including virtual and real images) taking into consideration the changes of the position of different objects in the environment, different lighting conditions and occlusions. The results show the effectiveness and the robustness of both methods.

Keywords: computer vision; position estimation; mapping; omnidirectional images; global appearance; Radon transform

1. Introduction

Nowadays, there are countless kinds of robots with innumerable configurations. Among them, mobile robots have been extended in recent years due to their flexibility, as they are able to change their position during operation. Usually, these robots have to solve a task in an unknown environment where the robot must estimate its position to be able to arrive at the target point, avoiding obstacles. There are two main approaches to solve the localization problem. First, the robot may have a map of the environment. In this case, the robot has to calculate its position within the map. Second, the robot may not have any *a priori* knowledge of the environment; thus, it has to create the map and then calculate its position. If these processes are carried out simultaneously, the problem is known as SLAM (simultaneous localization and mapping).

The robot can be equipped with many kinds of sensors to solve these problems, such as lasers, cameras, *etc*. They provide the robot with environmental information in different ways (e.g., lasers measure the distance to the nearest objects around the robot). This information is processed by the robot and permits building a map and estimating its position and orientation. With these data, the robot must be able to carry out its work autonomously.

Along the last few years, much research has been developed on robot mapping and localization using different kinds of sensors, and many algorithms have been proposed to solve these problems. Many these works use visual sensors, since they permit many possible configurations, and they provide the robot with very rich information from the environment. In this work, we consider a robot equipped with an omnidirectional vision system [1]. We can find many previous works that use omnidirectional images in mapping and localization tasks, such as [2–5]. Valiente *et al.* [2] present a comparison between two different visual SLAM methods using omnidirectional images. Maohai *et al.* [3] propose a topological navigation system using omnidirectional vision. Garcia *et al.* [4] present a map refinement framework that uses visual information to refine the topology of the environment. At last, Garcia *et al.* [5] make a survey of vision-based topological mapping and localization methods.

Traditionally, the developments in mobile robots using visual sensors are based on the extraction and description of some landmarks from the scenes, such as SIFT (scale-invariant feature transform) [6] and SURF (speeded-up robust features) [7] descriptors. This approach presents some disadvantages: the computational time to calculate and compare the descriptors is usually very high; thus, these descriptors may not be used in real time, and this leads to relatively complex mapping and localization algorithms. As an advantage, only a few positions need to be stored in the map to make the localization process possible.

More recently, some works proposed using the global information of the images to create the descriptors. These techniques have been demonstrated to be a good option to solve the localization and navigation problems in 2D. Chang *et al.* [8], Payá *et al.* [9] and Wu *et al.* [10] propose three examples of this approach. In [11], several methods to obtain global descriptors from panoramic scenes are analyzed and compared to prove their validity in map building and localization. The majority of these global appearance descriptors can be used in real time, because the computational time to calculate and handle them is low, and they usually lead to more straightforward mapping and localization algorithms.

In this work, we propose a solution to the mapping and localization problems using only the visual information captured by an omnidirectional vision system mounted on the robot. This system is composed of a camera pointing to a hyperbolic mirror and captures omnidirectional images of the environment. Each scene is described with a single global-appearance descriptor.

Our starting point is a database of omnidirectional images captured on a grid of points in the environment where the robot has to navigate.

Compared to previous works, the contributions of this paper are:

1. Describing the global appearance of a set of images: We develop a method based on the Radon transform [12] and the gist descriptor [13] to create a visual model of the environment. We have not found any previous work that uses the Radon transform as a global appearance descriptor to solve the mapping and localization problems in robotics.
2. Solving the mapping and localization problems with these descriptors: This contribution is two-fold:
    (a) Solving the localization problem with accuracy, with respect to a visual model (map) previously created: In this model, the positions where the images were captured are known.
    (b) Solving the localization problem with respect to a visual model where the capture positions of the images are not known: We have implemented a local mapping algorithm, and subsequently, we solve the localization problem.

The experiments have been carried out with several different image databases. The first one has been created synthetically from a virtual room. We also use some real databases to test the validity of our method.

The remainder of this paper is structured as follows. Section 2 introduces the concept of global appearance and the descriptors that we have developed. Section 3 presents the algorithm that we have implemented to solve the local mapping problem (spring method). Section 4 describes our localization method to estimate the position of the robot. In Section 5, the experiments and results are presented. At last, a discussion is made in Section 6, and Section 7 outlines the conclusions.

2. Global Appearance of Omnidirectional Images

Methods based on the global appearance of the scenes constitute a robust alternative compared to methods based on landmark extraction. The key is that the global appearance descriptors represent the environment through high level features that can be interpreted and handled easily, and with a reasonably low computational cost.

This section presents the transform that we have employed to describe the scenes (omnidirectional images). Each scene is represented through only one descriptor that contains information of the global appearance without any segmentation or local landmark extraction. We also present the distance measures we use to compare descriptors, and we study their properties.

Any new global appearance description method should satisfy some properties: (1) it should make a compression effect in the image information; (2) there should be a correspondence between the distance between two descriptors and the metric distance between the two positions where the images were captured; (3) the computational cost to calculate and compare them should be low, so that the approach can be used in real time; (4) it should provide robustness against noise, changes in lighting conditions, occlusions and changes in the position of some objects in the environment; and, at last, (5) it should contain information of the orientation that the robot had when it captured the image.

The description method we have employed is mainly based on the Radon transform. We also make use of the gist descriptor, because of its properties in localization tasks [11].

2.1. Radon Transform

The Radon transform was initially described in [12]. It has been used traditionally in some computer vision tasks, such as shape description and segmentation ([14,15]).

The Radon transform in 2D consists of the integral of a 2D function over straight lines (line-integral projections). This transform is invertible. The inverse Radon transform reconstructs an image from its line-integral projections. For this reason, it was initially used in medical imaging (such as CAT scan and magnetic resonance imaging (MRI)).

The Radon transform of a 2D function $f(i,j)$ can be defined mathematically as:

$$\mathcal{R}\{f(i,j)\} = \lambda_f(p,\phi) = \iint_{-\infty}^{+\infty} f(i,j)\,\delta(p - \vec{r}\,\hat{\vec{p}})\,di\,dj \tag{1}$$

where δ is the Dirac delta function ($\delta(x) = 1$ when $x = 0$, and $\delta(x) = 0$ elsewhere). The integration line is specified by the radial vector $\vec{p}$ that is defined by $\vec{p} = \hat{\vec{p}} \cdot p$, where $\hat{\vec{p}}$ is a unitary vector in the direction of $\vec{p}$. p is the $\vec{p}$ magnitude:

$$p = |\vec{p}| \tag{2}$$

The line-integral projections evaluated for each azimuth angle, ϕ, produce a 2D polar function, λ_f, that depends on the radial distance p and the azimuth angle ϕ. $\vec{r}$ is a cluster of points that is perpendicular to $\vec{p}$.

The Radon transform of an image $im(i,j)$ along the line $c_1(d,\phi)$ (Figure 1) can be expressed more clearly by the following equivalent expression:

$$\mathcal{R}\{im(i,j)\} = \int_{\mathbb{R}} im(i'\cos\phi - j'\sin\phi, i'\sin\phi + j'\cos\phi)\,ds \tag{3}$$

where:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} \tag{4}$$

When the Radon transform is applied to images, it calculates the image projections along the specified directions through a cluster of line integrals along parallel lines in these directions. The distance between the parallel lines is usually one pixel. Figure 2a shows the integration paths to calculate the Radon transform of an image in the ϕ direction, and Figure 2b shows the value of each component of the Radon transform in a simplified notation.

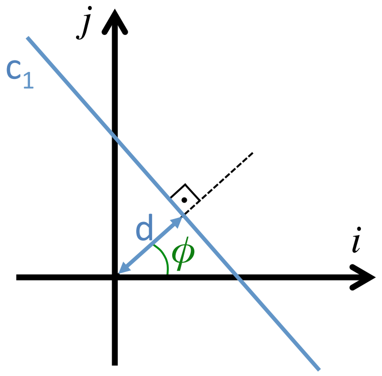
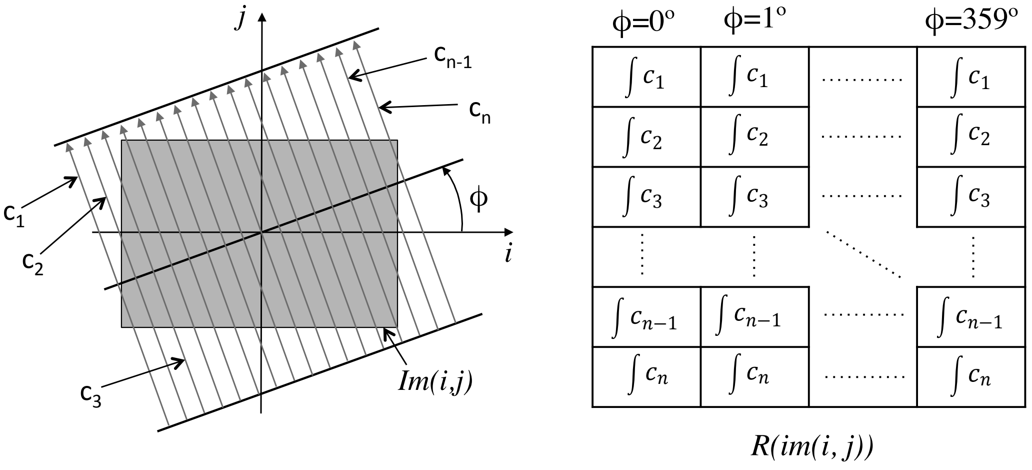


Figure 1. Line parametrization through the distance to the origin *d* and the angle between the normal line and the *i* axis, ϕ.

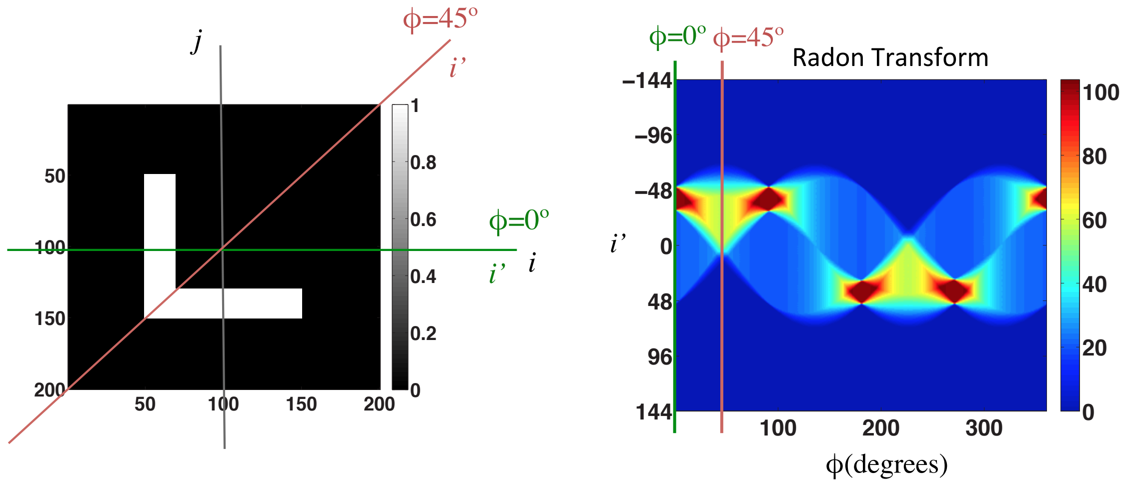


ϕ=0°	ϕ=1°		ϕ=359°
$\int c_1$	$\int c_1$	....	$\int c_1$
$\int c_2$	$\int c_2$	....	$\int c_2$
$\int c_3$	$\int c_3$	....	$\int c_3$
⋮	⋮	⋱	⋮
$\int c_{n-1}$	$\int c_{n-1}$	....	$\int c_{n-1}$
$\int c_n$	$\int c_n$	....	$\int c_n$

R(im(i, j))

(**a**) (**b**)

Figure 2. (**a**) Integration paths to calculate the Radon transform of the image $im(i, j)$ in the ϕ direction. (**b**) Radon transform matrix of the image $im(i, j)$.



(**a**) (**b**)

Figure 3. (**a**) Sample image. (**b**) Radon transform of the sample image.

Figure 3 shows a sample black and white image, on the left, and its Radon transform, on the right. Furthermore, it shows graphically the process to calculate the Radon transform.

2.1.1. Radon Transform Properties

The Radon transform has several properties that make it useful in localization tasks using omnidirectional images. These properties are the following:

- Linearity: The Radon transform meets the linearity property as the integration operation is a linear function of the integrand:
$$\mathcal{R}\{\alpha f+\beta g\}=\alpha\mathcal{R}\{f\}+\beta\mathcal{R}\{g\} \tag{5}$$
- Shift: The Radon transform is a variant operation to translation. A translation of the two-dimensional function by a vector $\overrightarrow{r_0}=(i_0,j_0)$ has a translation effect on each projection. This translation is given by a distance $\overrightarrow{r}\cdot(\cos\phi,\sin\phi)$.
- Rotation: If the image is rotated an angle ϕ_0, it implies a shift ϕ_0 of the Radon transform along the variable ϕ (columns shift).
- Scaling: A scaling of f by a factor b implies a scaling of the d coordinate and amplitude of the Radon transform by a factor b:
$$\mathcal{R}\left\{f\left(\frac{i}{b},\frac{j}{b}\right)\right\}=|b|\lambda_f\left(\frac{d}{b},\phi\right) \tag{6}$$

2.2. *Gist Descriptor*

The gist descriptors try to imitate the human perception system to describe the scenes. They identify regions with a prominent color or texture in relation to the environment. The gist concept was introduced by Oliva and Torralba [13] with the name holistic representation of the spatial envelope. The authors proved that this description method creates a multidimensional space in which the scenes with the same semantic category (e.g., streets, sky, buildings, *etc.*) are projected in near points.

Mathematically, the method tries to codify the spatial information by means of 2D Fourier transform in several regions of the image. These regions are distributed in a regularly-spaced grid. The set of descriptors in each region is dimensionally reduced by PCA (principal component analysis) to generate a unique scene descriptor with reduced dimension. In recent works, wavelet pyramids have been used instead of the Fourier transform, as in [16], where the authors carried out several experiments with different groups of natural images (coasts, forests, open fields, deserts, *etc.*) and artificial images (images of buildings, doors, interior of buildings, roads, *etc.*). The developed descriptors worked successfully to classify these images according to the above properties.

Recent works use the prominence concept together with gist, which highlights the zones that differ more from their neighbors [17]. This descriptor is built with the information of intensity, orientation and color.

In this work, some changes are made to this descriptor, since it will be used for a different purpose. We work with a set of indoor images with many similarities between them. The descriptor must be able to work with these images in such a way that the distance in the descriptor space reflects the geometrical distance between the points where the images were captured. Additionally, the Radon transform has been used instead of images. Since this transform can be interpreted as a grayscale image, only the concept of prominence is used (but not color information). The steps we have implemented to build the gist descriptor are:

1. Building a pyramid of Radon transforms: At first, a Gaussian pyramid of *n* Radon transforms is created to describe the Radon transform features in several scales. The first level of the pyramid is the original Radon transform. Each new level is obtained from the previous one, applying a Gaussian low-pass filter and subsampling it to reduce its resolution. Figure 4 shows an example of a Radon transform four-level pyramid.

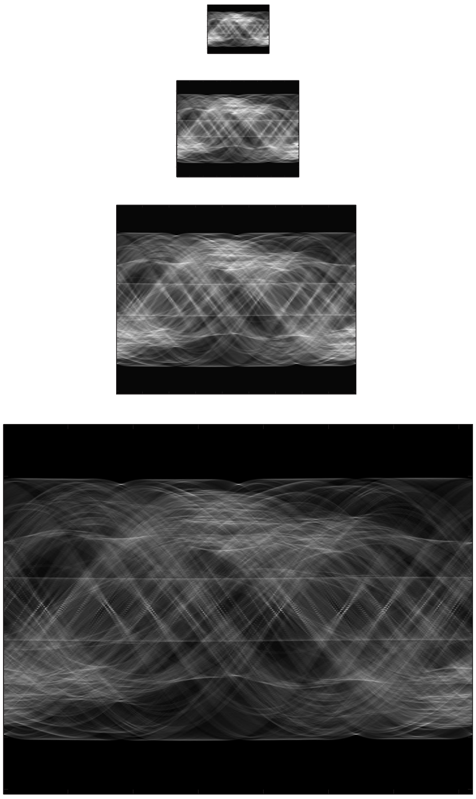


Figure 4. Example of a Radon transform four-level pyramid.

2. Gabor filtering: In order to incorporate the orientation information in the descriptor, each Radon transform of the two levels of the pyramid is filtered by four Gabor filters with different orientations $\theta = \{0, 45, 90, 135\}^\circ$. As a result, four matrices per pyramid level are obtained, with orientation information in the four analyzed directions.
3. Blockification. Finally, a dimensionality reduction process is needed. With this aim, we group the pixels of every matrix in blocks calculating the average intensity value that has the pixels in each block. We have decided to use horizontal blocks. This type of block is interesting, because the information in each block is independent of the robot orientation. The final descriptor will be composed of *b* horizontal blocks per pyramid level and Gabor filter, since it will be used only for localization purposes (additionally, we could have defined vertical blocks if we had needed to compute the robot orientation [11]).

Figure 5 shows the process to obtain the gist descriptor. In the Figure 5a, we can observe the Radon transform of an omnidirectional image. The use of each Gabor filter in the Radon transform and the blockification is shown in Figure 5b. In this case, $n = 2$ levels, and $b = 8$ horizontal blocks. Finally, in Figure 5c, the final composition of the gist descriptor is shown. The size of the descriptor is equal to $4 \cdot n \cdot b$ components.

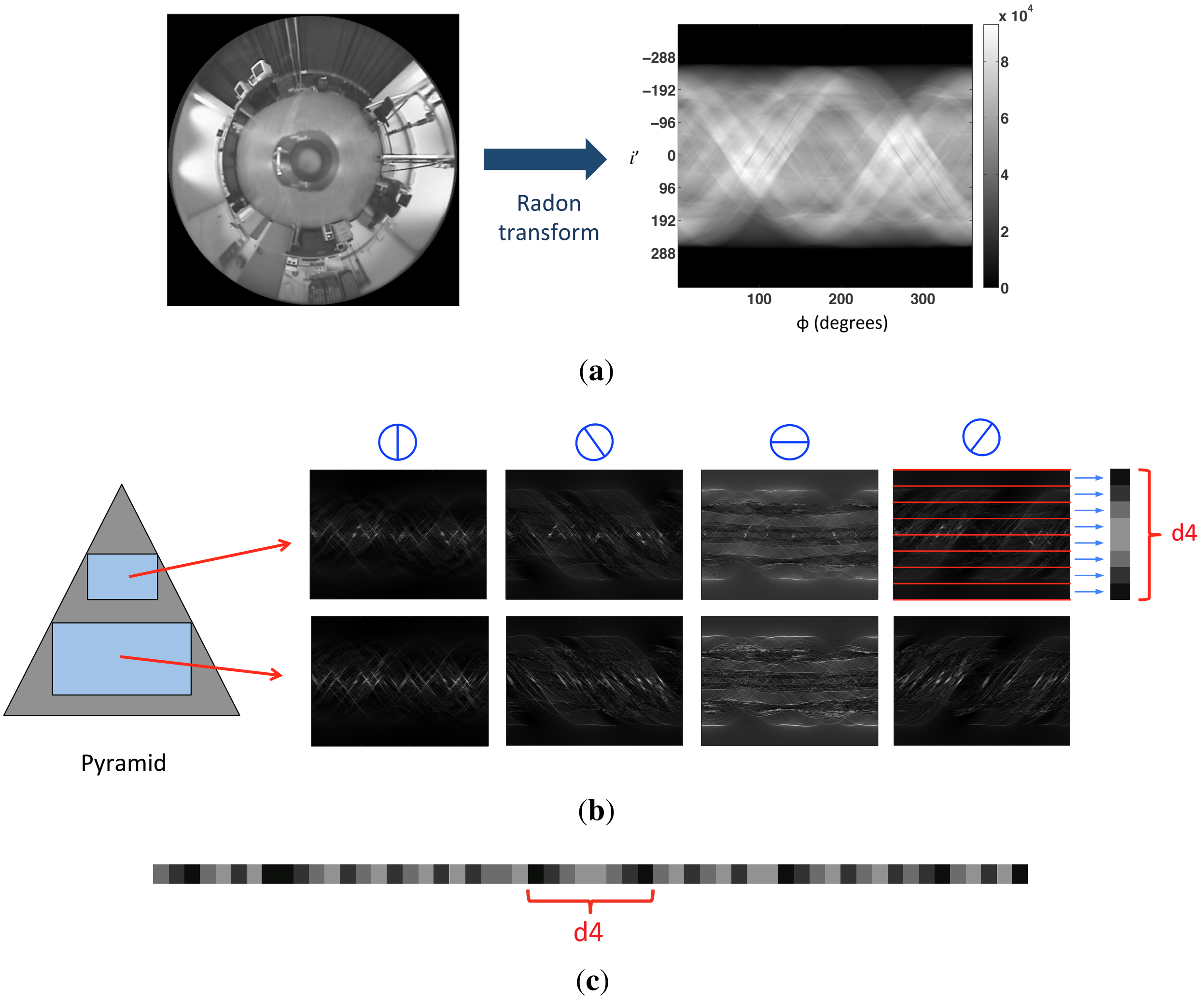


Figure 5. (**a**) The Radon transform of an omnidirectional image; (**b**) Matrices obtained after applying each of the four Gabor filters to the two levels of the pyramid; (**c**) Final composition of the gist descriptor.

2.3. Phase-Only Correlation

In this subsection, we present the method employed to compare the Radon transform of two images.

In general, a function in the frequency domain is defined by its magnitude and its phase. Sometimes, only the magnitude is taken into account, and the phase information is usually discarded. However, when the magnitude and the phase features are examined in the Fourier domain, it follows that the phase features contain also important information, because they reflect the characteristics of patterns in the images.

Oppenheim and Lim [18] have demonstrated this by reconstructing images using the full information from the phase with unit magnitude. This shows that the images resemble the originals, in contrast to reconstructing images using the full information from the magnitude with uniform phase.

Phase-only correlation (POC), proposed in [19], is an operation made in the frequency domain that provides a correlation coefficient between two images [20]. In our case, the objective is to compare the Radon transforms of two different images. This does not affect the POC performance, because the Radon transform can be interpreted as an image. Therefore, we explain POC using images.

The correlation between two images $im_1(i,j)$ and $im_2(i,j)$ calculated by POC is given by the following equation:

$$C(i,j) = \mathcal{F}^{-1}\left\{ \frac{\mathbf{IM}_1(u,v)\cdot\mathbf{IM}_2^*(u,v)}{|\mathbf{IM}_1(u,v)\cdot\mathbf{IM}_2^*(u,v)|} \right\} \tag{7}$$

where $\mathbf{IM}_1$ is the Fourier transform of Image 1 and $\mathbf{IM}_2^*$ is the conjugate of the Fourier transform of Image 2. $\mathcal{F}^{-1}$ is the inverse Fourier transform operator.

To measure the similitude between the two images, the following coefficient is used:

$$sim_{POC}(im_1, im_2) = max\{(C(i,j)\} \tag{8}$$

This coefficient takes values in the interval $[0,1]$, and it is invariant under shifts in the i and j axes of the images. Furthermore, it is possible to estimate these shifts Δ_i and Δ_j along both axes by:

$$(\Delta_i, \Delta_j) = argmax_{(i,j)}\{C(i,j)\} \tag{9}$$

If we use Radon transforms instead of images, the value Δ_i allows us to estimate the change of the robot orientation when capturing the two images.

This way, POC is able to compare two images independently on the orientation, and it is also able to estimate this change in orientation.

2.4. Distance Measure

To carry out the localization process, we need any mechanism to compare descriptors (distance measure). We have tested two different methods.

First, we can measure the distance between the Radon transform (RT) of two scenes by means of the POC magnitude (POC distance) by the following equation:

$$dist_{POC}(RT_1, RT_2) = 1 - sim_{POC}(RT_1, RT_2) \tag{10}$$

Second, we can make use of the gist descriptor to obtain a distance measure between images following the next steps:

1. Firstly, the two omnidirectional images are transformed by the Radon transform to create two descriptors RT_1 and RT_2.
2. Secondly, RT_1 and RT_2 are transformed with the gist approach (Section 2.2) to obtain g_1 and g_2.
3. Finally, the Euclidean distance between g_1 and g_2 is obtained (gist distance).

Figure 6 shows the POC and the gist distance *vs.* the geometric distance between the points where the images were captured. According to this figure, the POC distance shows a nonlinear behavior. This method seems to be a promising option to identify the nearest image, but it is not good to estimate the distance between images. The POC distance can be linearized by the following expression:

$$dist_{linearized} = dist^2_{POC} \tag{11}$$

where $dist_{linearized}$ is the final distance and $dist_{POC}$ the original POC distance. Figure 7 shows the POC distance and the linearized POC distance *vs.* the geometric distance of one aleatory linear path using our virtual database.

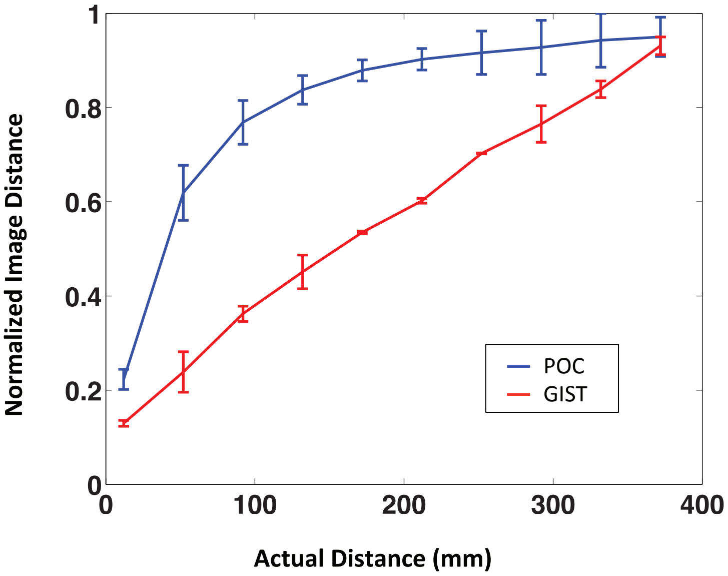


Figure 6. POC distance and gist distance *vs.* geometric distance. Each distance measure has been calculated along four different paths.

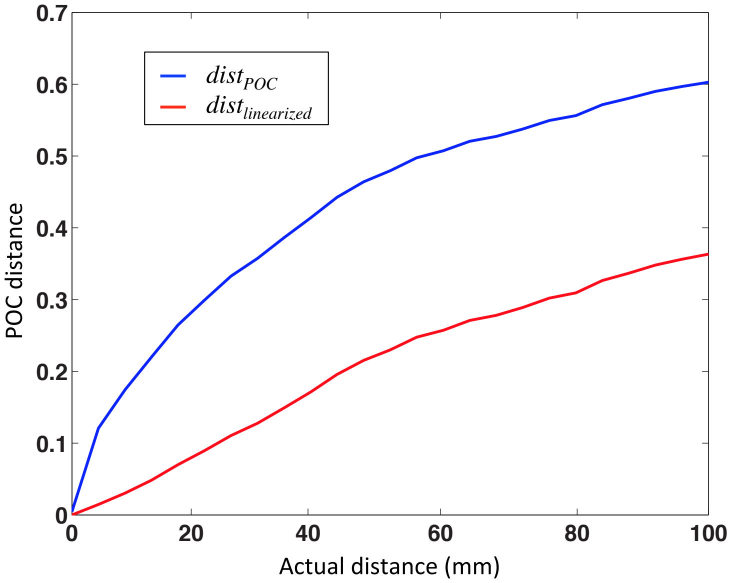


Figure 7. Phase-only correlation (POC) distance and POC distance linearized *vs.* geometric distance.

On the other hand, Figure 6 shows that the gist distance presents a quite linear behavior, so we expect it to be useful in mapping tasks.

3. Local Map Building (Spring Method)

In this section, we address the problem of building local maps of an unknown environment. We start from a set of omnidirectional images captured from different points of the environment to map, and we have no information of the capture positions nor of the order they were captured. Our objective is to estimate these positions in local regions of the environment. Thanks to this, we will be able to carry out a subsequent localization process (Section 4).

The mapping problem is solved using a mass-spring model, which represents the final map as a graph formed by nodes linked through connectors. Each node contains one image, and the connectors represent the neighbor relationships between two nodes, so that the images that have been captured sequentially are expected to belong to nodes that are connected together.

The only information we use to build the map is the distance among image descriptors. Once the map is built, we make use of the Procrustes transformation [21] to compare the resulting map with the actual map.

The method is based on a physical system of forces named the spring model [22,23]. The model is formed by the combination of two principles: Hooke's law and Newton's second law. Figure 8 shows the physical model scheme using three nodes. Each node N_i corresponds to an image, and they are linked with the springs S_{ij} whose natural length l_{ij0} is the distance between descriptors of images i and j. This method connects all of the nodes that have been previously chosen to create the local map and estimate the accuracy of the process.

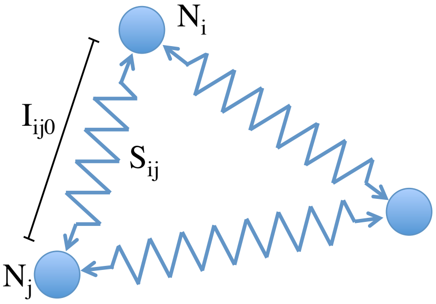


Figure 8. Spring model.

This method is based on the calculation of the elastic forces exerted by the springs on each node. Once the resulting force exerted on each node has been calculated, we proceed to obtain the acceleration, speed and position of each node.

The mapping process consists of solving at each time instant the next equations:

$$\overrightarrow{F_i} = \sum_{S_{ij} \in S} \left(-K_{ij}\left(l_{ij0} - l_{ij}\right) - b_{ij}\left(v_i - v_j\right)\right) \tag{12}$$

$$\overrightarrow{a_i} = \frac{\overrightarrow{F_i}}{m_i} \tag{13}$$

$$\overrightarrow{v_i}(t+\Delta T) = \overrightarrow{v_i}(t) + \overrightarrow{a_i}(t)\cdot\Delta T \tag{14}$$

$$\overrightarrow{r_i}(t+\Delta T) = \overrightarrow{r_i}(t) + \overrightarrow{v_i}(t)\cdot\Delta T, \tag{15}$$

where Equation (12) corresponds to Hooke's harmonic oscillator law. The resulting force on node i, $\overrightarrow{F_i}$, depends on the length of the springs l_{ij} and the elastic constant of the spring K_{ij}. A damper with damping constant b_{ij} has also been added between nodes, as it helps to improve the convergence of the algorithm. Newton's second law allows us to obtain the acceleration $\overrightarrow{a_i}$ using the node mass m_i. To simplify the system of forces, we used a mass equal to one for all nodes. Finally, the last two expressions are used to update the calculation of the speed $\overrightarrow{v_i}$ and the position $\overrightarrow{r_i}$ of all nodes in the system at each iteration (for each time increment, ΔT).

4. Localization Methods

In this section, we address the localization problem. Initially, the robot has a map of the environment. It is composed of a set of omnidirectional images of this environment. Then, the robot captures an image from an unknown position (test image). Comparing this image to the visual information stored in the map, the robot must be able to estimate its position.

To develop the algorithms, we take into consideration two different situations: (1) the positions where the map images were captured are known (pure localization); and (2) these positions are not known, and they must be estimated prior to the localization process (local map building and localization).

We have designed these algorithms to solve the problems:

1. Detecting the nearest image of the map.
2. Local mapping and localization.

The first method allows us to know the nearest image of the map (the most similar to the test image). Thanks to this information, we know that the robot is located in the surroundings of the corresponding image. The second method consists of calculating the image distance between the nearest images of the map and the test image. The number of nearest images is a parameter that can be modified to optimize the method. With these distances, we use the spring method to estimate the position of each image. As a result, we have the local map and the localization of the robot. These two methods are detailed in the following subsections.

4.1. Nearest Position of the Map

The operation consists of the following steps:

1. The robot captures an omnidirectional image from its current unknown position (test image). The objective is to estimate this position.
2. This image is transformed using the Radon transform.

3. The Radon transform of the test image is compared to all of the Radon transforms of the map using the POC comparison. As a result, we know which is the most similar image in the map to the test image.
4. The position where this omnidirectional image was taken is the nearest neighbor.

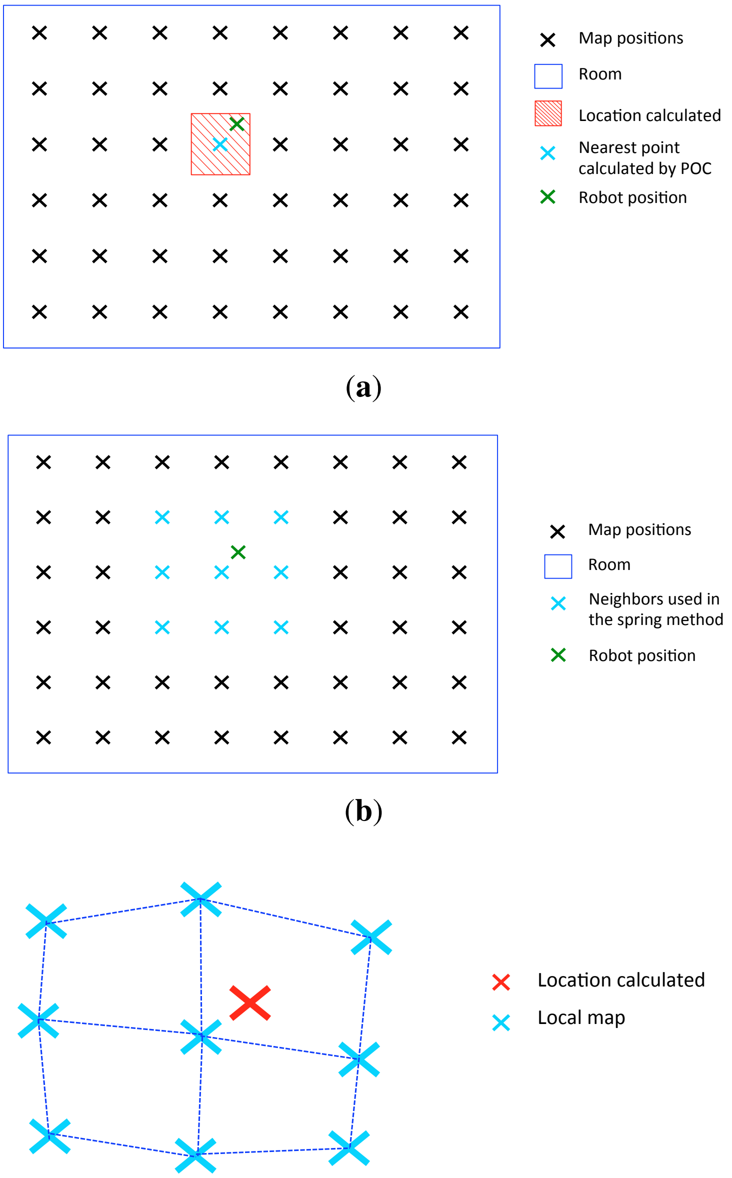
After this process, we assume that the robot is around this position. This is an absolute localization process, since we did not know the previous position of the robot nor its path. The accuracy depends mainly on the distance between the images of the map.

4.2. Local Mapping and Localization

In this subsection, we propose a method to localize the robot without any information of the position where the images of the map were captured. We have the Radon transforms of each image, but we do not have information about the position in which these images were captured. This localization method consists of the following steps:

1. We choose a number of nearest neighbors to use. These nearest neighbors are the descriptors that present the lowest POC distance to the Radon transform of the test image.
2. The gist descriptor of each Radon transform is calculated including the test image.
3. The distance between each pair of gist descriptors is obtained. We test two different distance measures: the Euclidean and the POC distance.
4. The spring method is used with these distances. This step provides a relative position between each gist descriptor. As a result, the position of the map images (local map) and the position of the test image are now known. This is our estimation of the robot position.
5. To estimate the localization error with respect to the local map, we must take into account that this map may present a rotation and a scale factor comparing to the actual position of the images. To consider these effects, we use a Procrustes approach [21] to estimate the error. This method removes the rotation and scale factor of the local map. The final error is the distance between the actual position of the robot and the calculated position after removing both effects.

Figure 9a shows the location area calculated by the first method (nearest position of the map). Figure 9b shows the location calculated by the second method with the nine nearest neighbors (local mapping and localization). In this figure, the map positions are represented at the actual capture points for representation purposes, but the algorithm does not know these positions. In Figure 9c, the local map created and the localization of the robot is shown. To make this local map, the initial position of each node to launch the spring method is random. This randomization may have an effect on the final result. For this reason, we run the algorithm several times. The results of this spring method are the local map and the robot localization.



(**a**)

(**b**)

(**c**)

Figure 9. (**a**) Nearest position of the map; (**b**) Map positions and neighbors used in the spring method; (**c**) Local mapping and localization.

5. Experiments and Results

In this section, first, we present the virtual database created to test the methods and the results obtained with it. Second, we describe the database of real images that we have used and the results obtained with this second database.

5.1. Virtual Database

In order to check the performance of the proposed technique, we have created a virtual environment that represents an indoor room. In this environment, it is possible to create omnidirectional images from any position using the catadioptric system showed in Figure 10. Figure 11a shows a bird's eye view of the environment.

The omnidirectional images have 250 × 250 pixels, and they have been created using a catadioptric system composed of a camera and a hyperbolic mirror whose geometry is described in Figure 10. The parameters used in the mirror equation are $a = 40$ and $b = 160$.

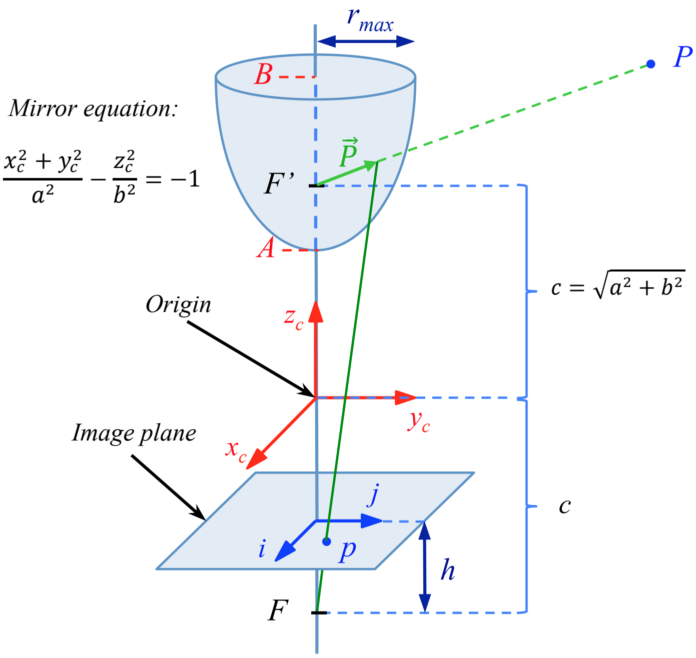
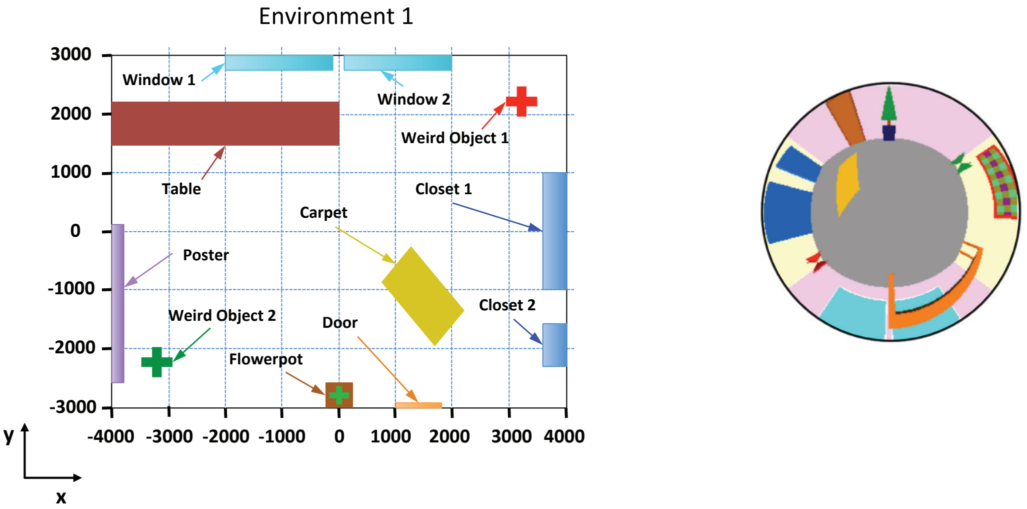


Figure 10. Catadioptric system used to capture the synthetic omnidirectional images.



(**a**) (**b**)

Figure 11. (**a**) Bird eye's view of the virtual environment; (**b**) Example of an omnidirectional image captured from the points x = 0 and y = 0.

Several images have been captured in the environment to create the map from several positions on the floor. The map is composed of 4800 images captured on a 8 × 6 meter grid with a step of 10 cm between images. To carry out the experiments, we can change the step of the grid to test its influence. We must take into account that the higher the grid step, the fewer images compose the map. Figure 11b shows one sample omnidirectional image of the environment created with our program.

5.2. Results Obtained with the Virtual Database

To test our methods, we will study the influence of some parameters, such as the grid step of the map and the number of neighbors used in the spring method. In this subsection, we compare the results of the tests to try to optimize each method.

Firstly, we analyze the first method (obtaining the nearest position of the map) and the influence of the grid step of the map. Secondly, we analyze the second method (local mapping and localization) and the influence of the grid step of the map and the number of nearest neighbors.

5.2.1. First Method: Detecting the Nearest Image of the Map

In this experiment, we analyze four different step sizes between consecutive map positions. The distances that we will use are 100, 200, 300 and 400 mm. The size of the grid is 8 m × 6 m in all cases, so the number of map images depends on the step size. It will have an important influence on the computational cost.

Figure 12 shows the POC distance (Equation (10)) between the Radon transform of a test image and each image of the map (200 × 200). The position of the test image is x = -2239 mm and y = -1653 mm. Additionally, the corresponding position according to this figure (the minimum of the 2D function) is x = -2200 and y = -1600. As we can see, the distance decreases sharply around this position.

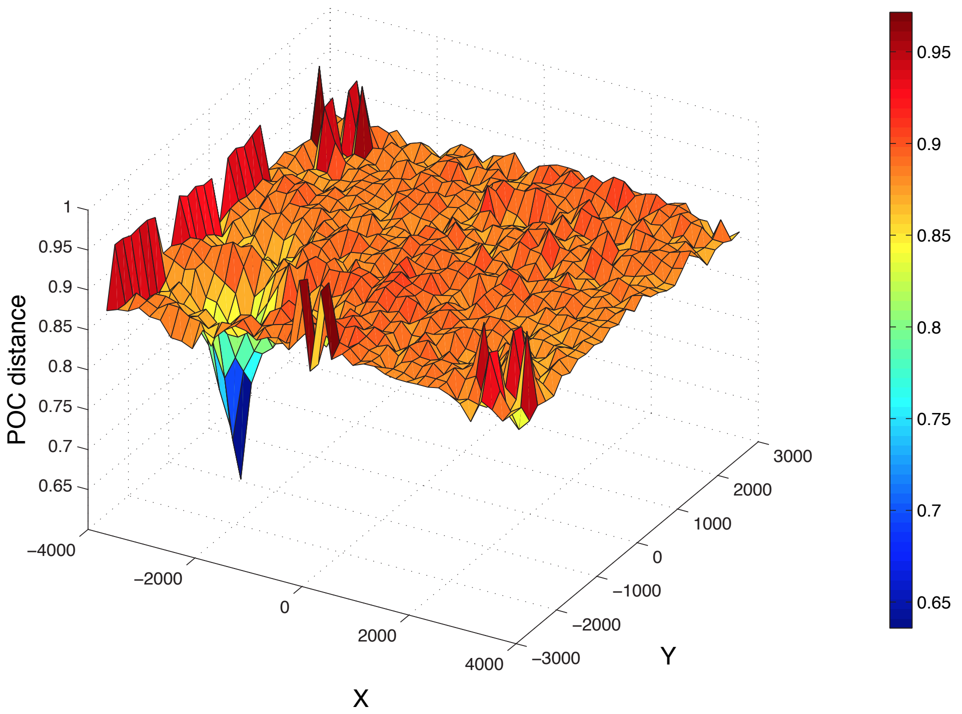


Figure 12. Distance between the Radon transform of a sample test image and all of the images of the map. The position of the test image is x = -2239 mm and y = -1653 mm.

Figure 13 shows the final result of this test. We have used 3500 test images captured from different random positions of the environment. In each bar, the blue part represents the proportion of correct localizations; the green part represents that the method has localized the second nearest position (*i.e.,* the position calculated is not the nearest position, but rather the second nearest position of the map); and the red part is the proportion of errors for each map size.

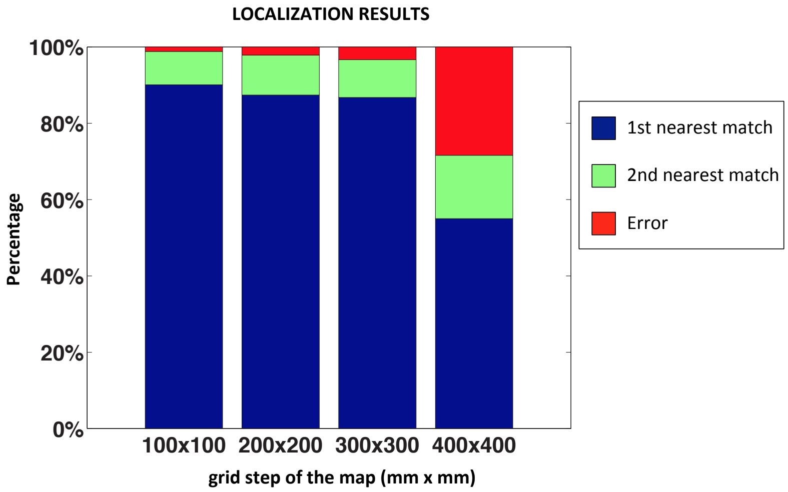


Figure 13. Results of the nearest image experiments: success rate.

We can observe that the method is working properly with a distance between map positions not exceeding 300 × 300 millimeters, because if this value increases, the POC distance begins to saturate, as stated in Section 2.4.

5.2.2. Second Method: Local Mapping and Localization

In this subsection, the second method is analyzed, and we study the influence of the distance between map positions and the number of nearest neighbors used in the spring algorithm. To do this experiment, we have chosen the same distances between map positions as in the previous subsection: 100, 200, 300 and 400 mm. Then, we will choose the value that provides the best result, and we will then optimize the number of nearest neighbors: we will test grids of 9, 25 and 49 nearest neighbors.

The algorithm has been launched with 500 test images captured from random positions generated with our program, repeating the experiment four times with each test image. Figure 14 shows the error distribution of the experiments with the virtual database using the POC distance, depending on the number of nearest neighbors and the grid step of the map. This error is the localization error of the test image calculated according to Step 5 (Section 4.2). Figure 15 presents the error distribution of the same experiments, but with the gist distance. In this environment, the gist distance presents a lower localization error than the POC distance. The parameters used in the gist descriptors for the experiments are $n = 2$ levels and $b = 16$ horizontal blocks.

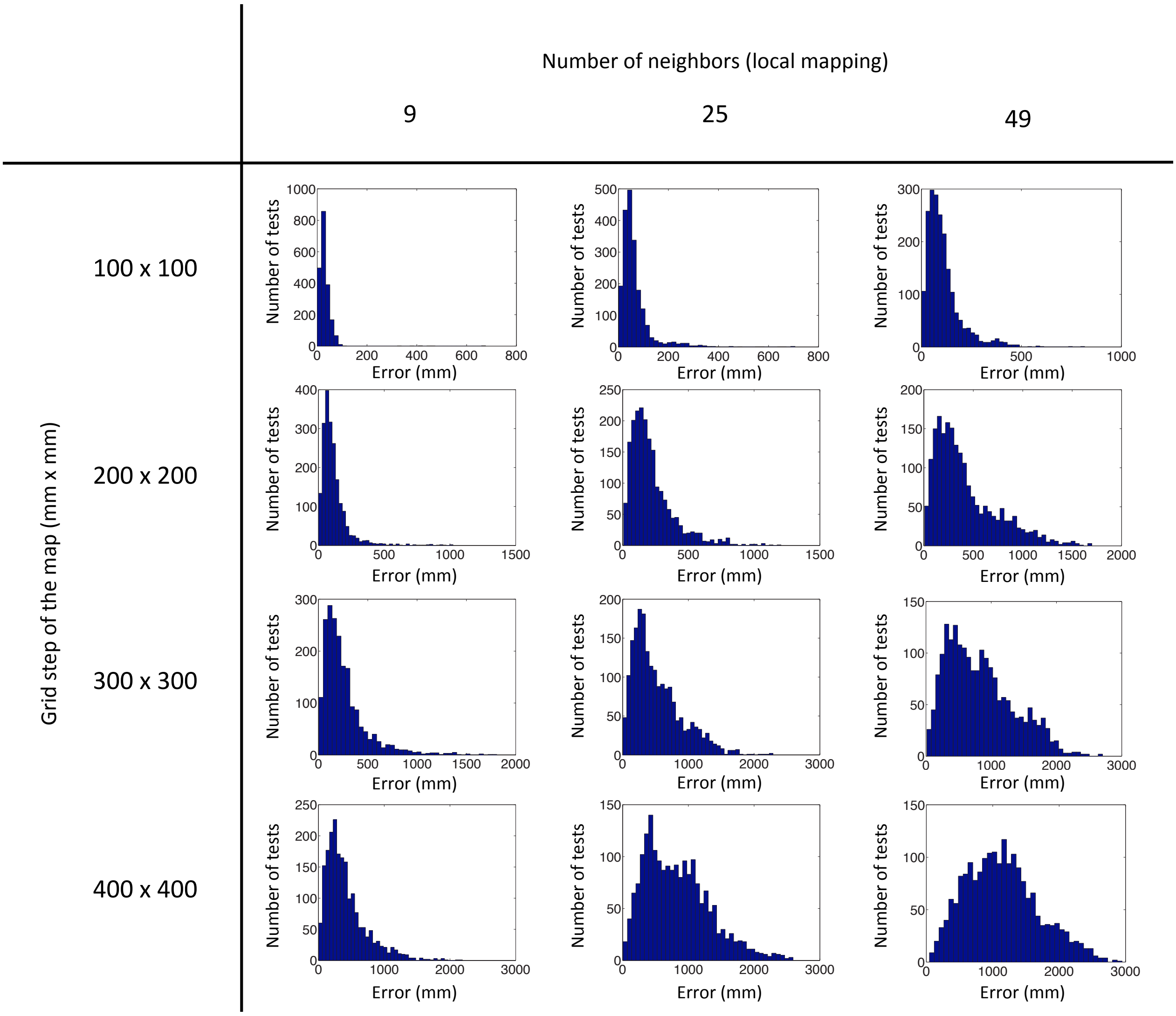


Figure 14. Error distribution of the experiments with our virtual database, using POC distance.

As we can see in the figures, the best results of the test have been achieved with a grid step equal to 100 mm and 200 mm, but we must also consider the computation time. Figure 16 shows the average computation time per iteration in each case, in the case that we have chosen 25 nearest neighbors. The distance between map positions of 100 mm is not advisable, because the computation time is relatively high. For this reason, we choose the distance of 200 mm to do the following tests, with actual images, because taking into account the error and the computation time, it is the best option. We can observe that the computational time of the second method is the same for all cases, because it does not depend on the size of the map, but rather on the number of nearest neighbors.

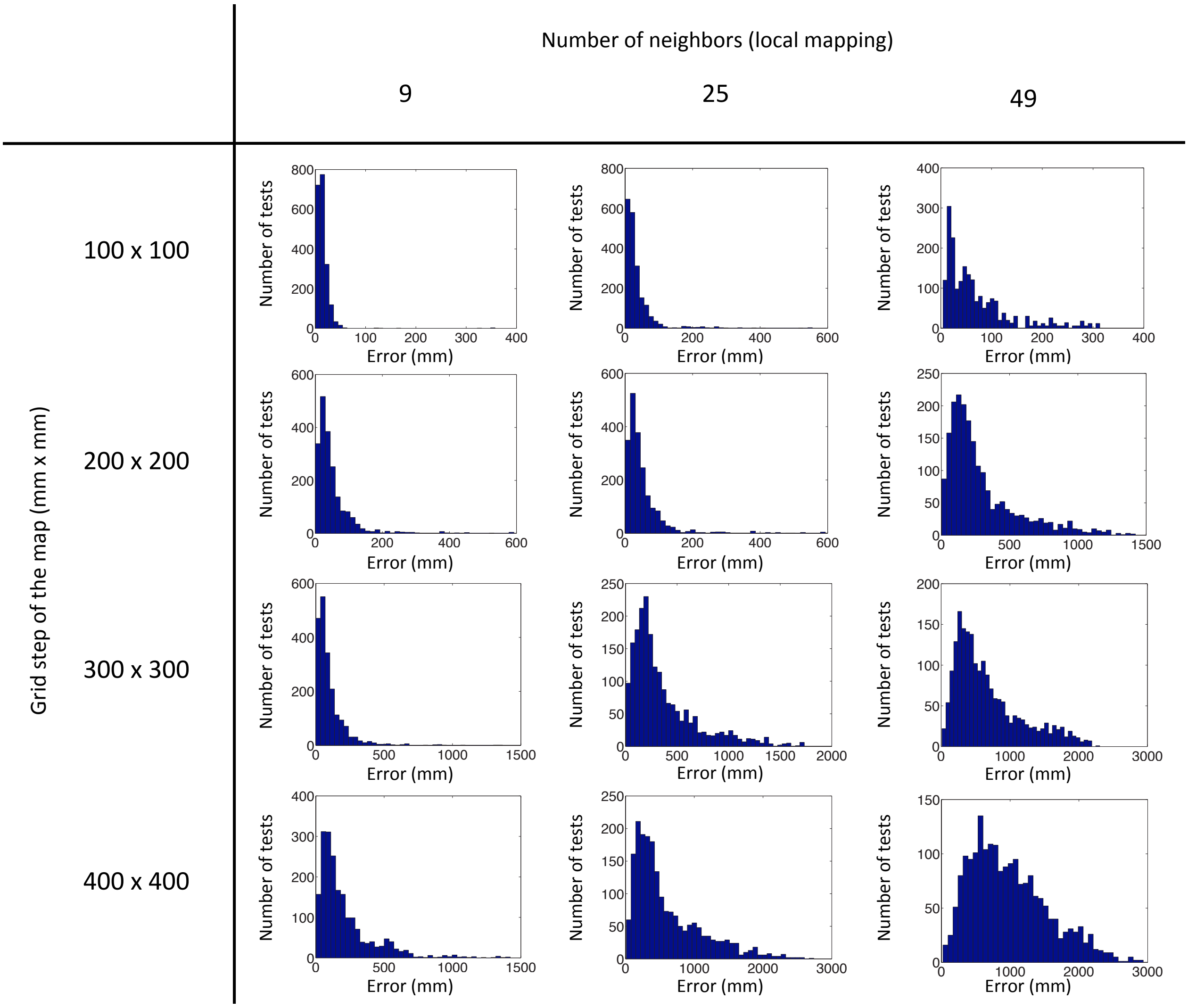


Figure 15. Error distribution of the experiments with our virtual database, using gist distance.

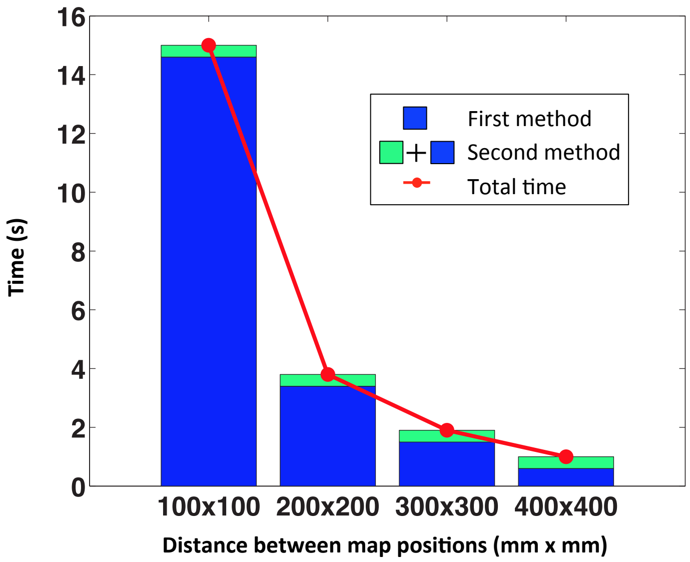


Figure 16. Computational time for each iteration. Using 25 nearest neighbors in the second method.

5.3. Results with Actual Databases

Once we have analyzed our methods with a virtual database, now we will test them with some databases composed of images captured in real working environments under real lighting conditions.

First, we test the first method using a third-party database that has different lighting conditions.

Second, we test the second method with our own databases (two different rooms of our university).

5.3.1. First Method: Detecting the Nearest Image of the Map

To test this method, we have used the Bielefeld database [24]. This database contains several image maps with changes in the position of some objects and in the lighting conditions. The images were captured on a 10 × 17 grid with a 30-cm step. Hence, the area of the capture grid was 2.7 m × 4.8 m, which covered nearly all of the floor's free space. These image maps are:

- Original: the standard or default condition of the room. Images were collected with both overhead fluorescent light bars on, the curtains and door closed and no extraneous objects present.
- Chairs: three additional office chairs were positioned within the capture grid.
- Arboreal: a tall (3 m) indoor plant was added to the center of the capture grid.
- Twilight: curtains and door open. The image collection lasted from sunset to the evening.
- Doorlit: the light bar near the window was switched off.
- Winlit: the light bar near the door was switched off.

To carry out this first experiment, we have used the "original" set as our map of images and the images in the other sets as test images. Figure 17 shows the success rate of this experiment. We have used 20 random test images of each database to obtain this figure.

This is quite an interesting experiment, as it shows the robustness of the algorithm, even when the test images present changes with respect to the map.

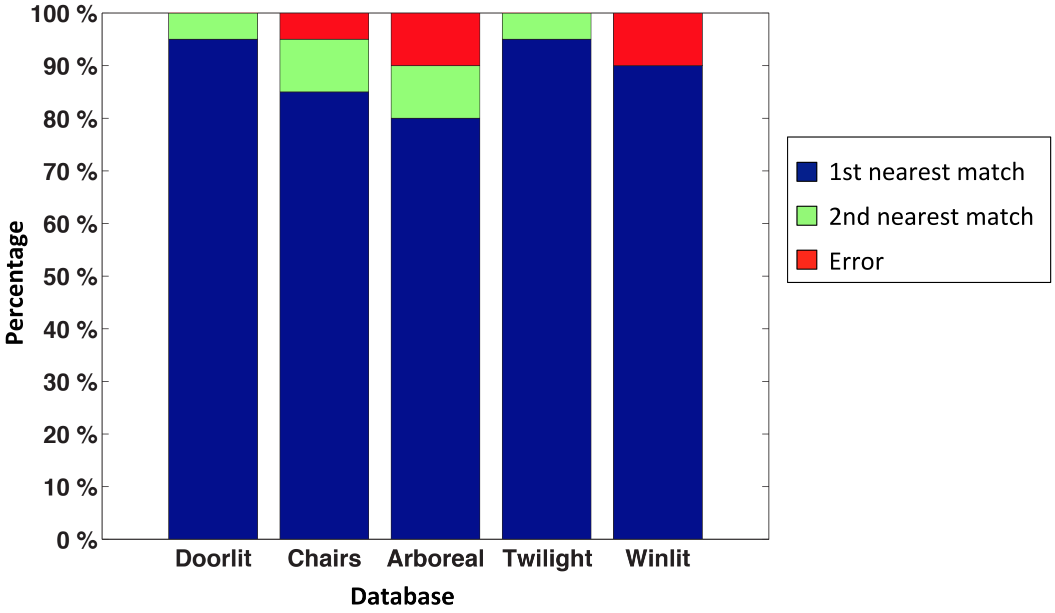
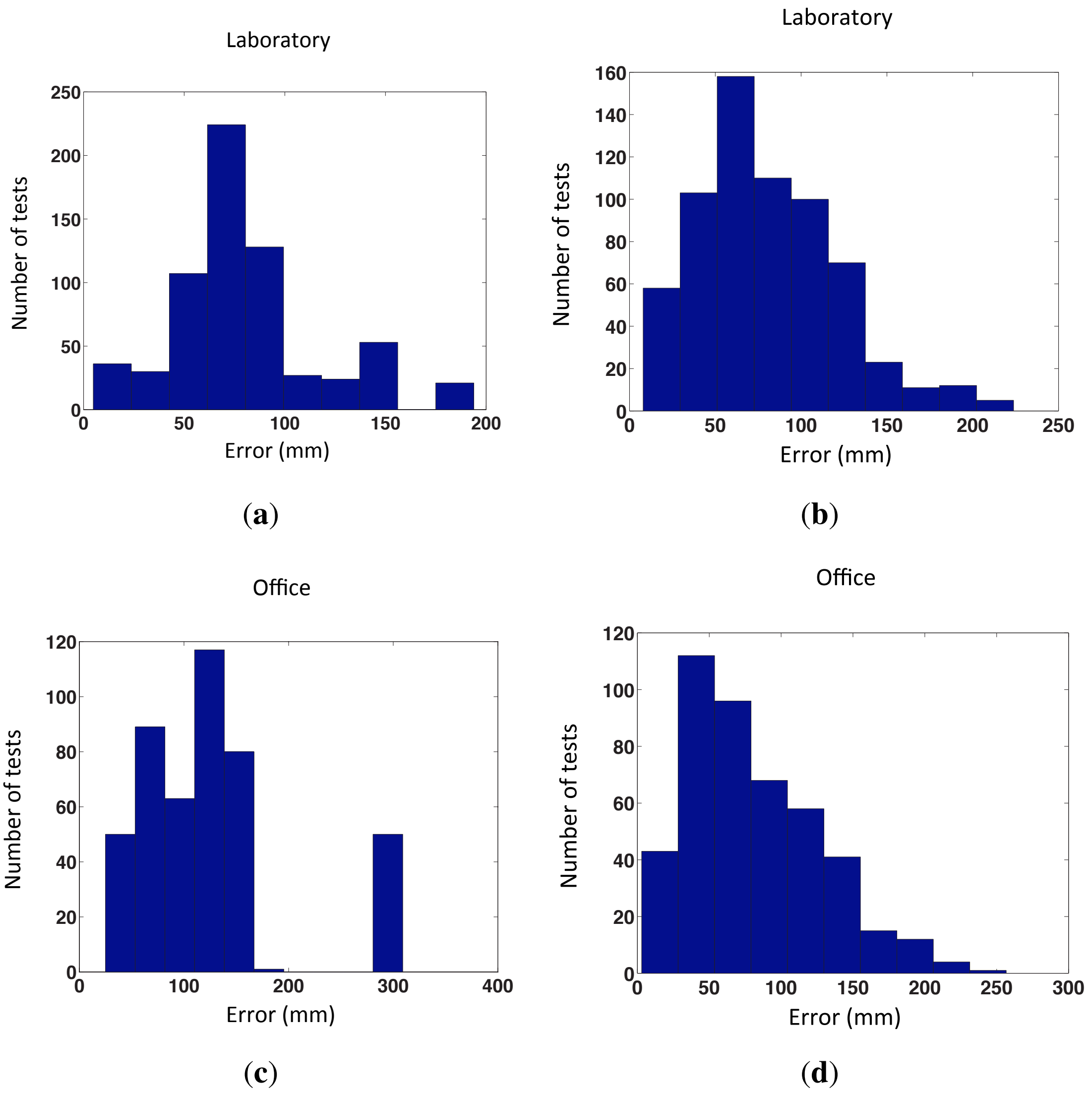


Figure 17. Success rate with the Bielefeld database.

5.3.2. Second Method: Local Mapping and Localization

In this case, we have used our database of images, captured at our university. It covers two different rooms, a laboratory and an office. The images have been captured on a grid with 8 × 8 images with a 20-cm step, in both cases.

We follow two steps. First, we detect the nearest neighbors by the POC comparison of the Radon transforms of the map images. Second, we create the local map with these Radon transforms using the gist or POC distance. Finally, we localize our robot in this local map. Figure 18 shows the error distribution of the second method using our databases ("laboratory" and "office") and both distance measures (POC and gist distance). The experiment has been carried out with 13 test images and 50 times per test image in the laboratory and with mine test images and 50 times per test image in the office. Figure 19 shows the error distribution of the second method using our databases, adding occlusions to each test image. Two examples of omnidirectional images with occlusions are shown in Figure 20.



(**a**) (**b**)

(**c**) (**d**)

Figure 18. Error distribution of the second method. (**a**,**b**) Laboratory database and gist and POC distances, respectively; (**c**,**d**) Office database and gist and POC distances, respectively. The distance between map positions in the four experiments is 200 × 200 mm.

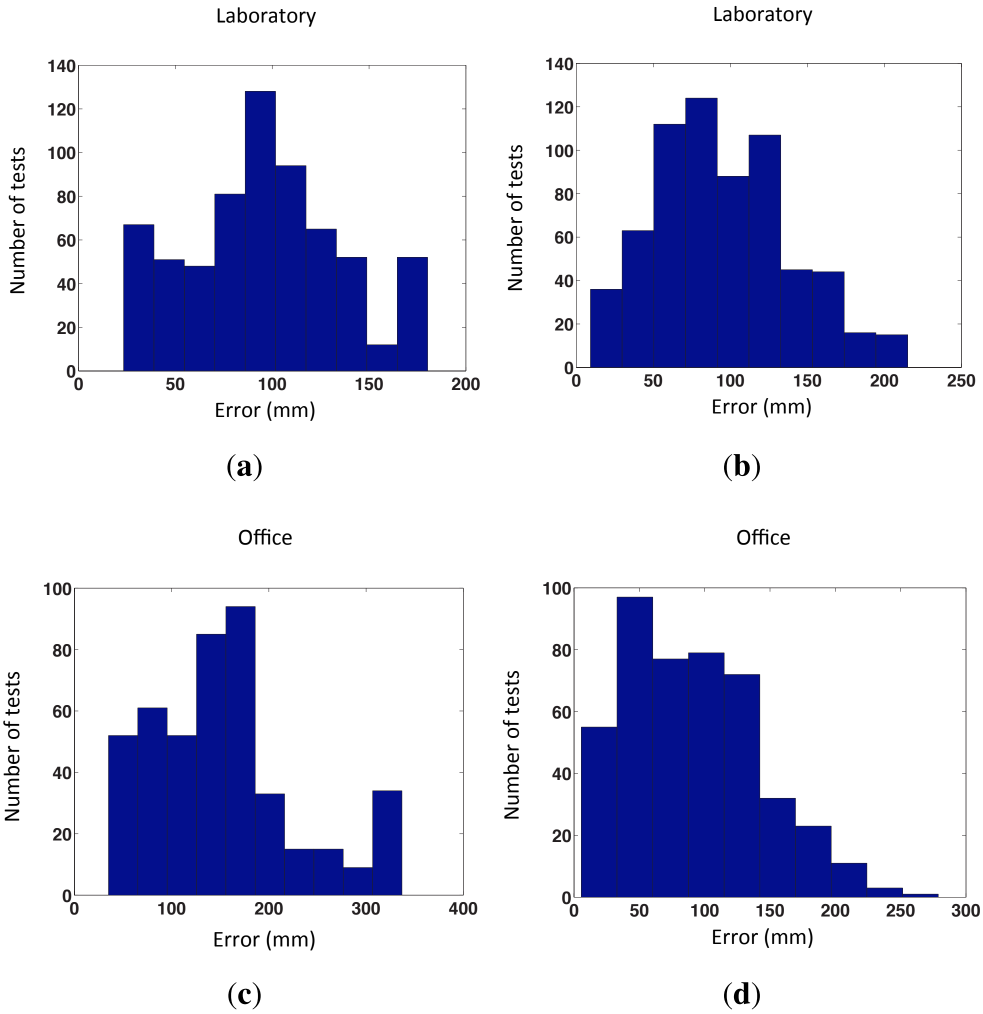


Figure 19. This figure represent the same experiments of Figure 18, adding random occlusions to the test image. (**a**,**b**) Laboratory database and gist and POC distances, respectively; (**c**,**d**) Office database and gist and POC distances, respectively. The distance between map positions in the four experiments is 200 × 200 mm.

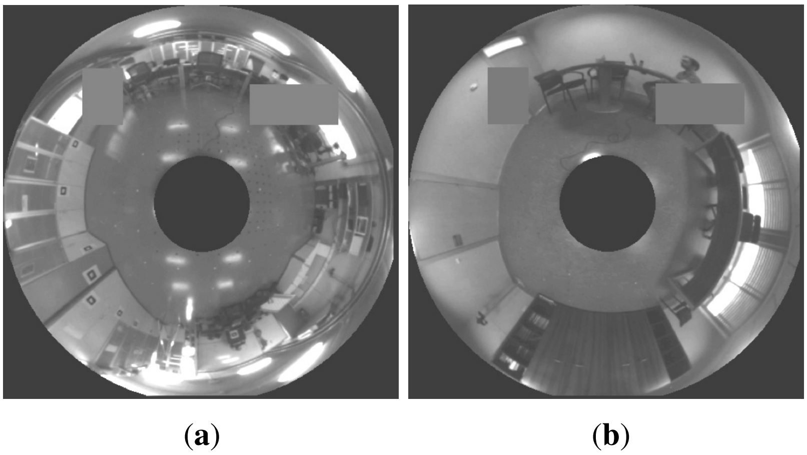


Figure 20. (**a**) Example of an omnidirectional image captured in our laboratory with two occlusions; (**b**) Example of an omnidirectional image captured in our office with two occlusions.

6. Discussion

Once we have presented the results, in this section, we provide a discussion of these results with both methods. We have arrived at some conclusions about the advantages and disadvantages of each method with different values of the variable parameters.

The first method (detecting the nearest image of the map) is a very good method if the robot has information of the positions in which the map images were taken. It permits carrying out an absolute localization, because the robot needs no information of its previous location. The results have demonstrated that it is a very reliable method to estimate the nearest position of the map, even when some changes are present (lighting conditions, new objects, *etc.*). As for the values of the parameters, the distance between map positions is the variable parameter in this method. The best choice is 200 × 200 mm, as it offers a good balance between accuracy and computational time.

The second method (local mapping and localization) is a good method if the robot does not know the map positions, because it creates a local map. This method also localizes the nearest images and the test image within this local map. As for the values of the parameters, first, the best choice is a distance between map positions of 200 × 200 mm, as it offers a good balance between accuracy and computational time; second, the best number of nearest neighbors is nine, because it leads to lower error; and finally, the kind of distance (POC or gist distance) should be chosen depending on the characteristics of the environment, because with the virtual database, the gist distance is better, but with the actual database of our university, the POC distance has presented more accurate results.

To finish, a comparative evaluation has been carried out between the proposed descriptor (Radon) and two other classical description methods in a localization task. The objective is to study the performance of the proposed descriptor, both in terms of localization accuracy and computational cost.

The description methods we use to make this comparison are the Fourier signature (FS) [11], a classical method to describe the global appearance of a panoramic scene, and SIFT [6], a typical method to extract and describe local features from a scene. The test consists of determining the nearest image from a set of omnidirectional images, and the database used is the laboratory (from the actual database captured at our university).

First, an experiment has been carried out to optimize the computational cost of the Radon transform (RT) method. With this aim, we repeat the localization experiment several times, using different resolutions, from RT $\in \mathbb{R}^{709x360}$ to RT $\in \mathbb{R}^{22x6}$. The results (success rate and computational cost) are shown on Figure 21. As shown, RT resolution can be reduced to 173 × 45 keeping the success rate, which clearly improves the computational cost of the process. This way, we choose this resolution to carry out the comparison.

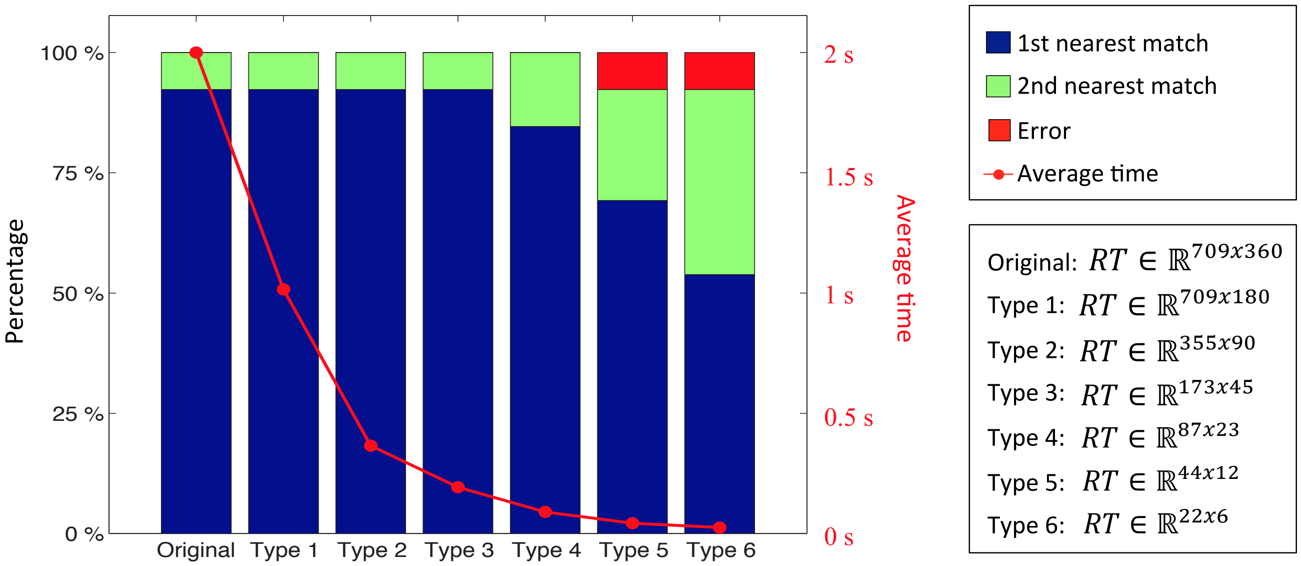


Figure 21. Success rate and average time using different types of Radon transforms with the laboratory database.

When using the Fourier signature, each omnidirectional scene is transformed to the panoramic format and described according to [11]. The Euclidean distance is used to compare two descriptors. On the other hand, when using SIFT, the SIFT features of the omnidirectional scenes are extracted, described and compared. The distance between two images im_1 and im_2 is defined in Equation (16).

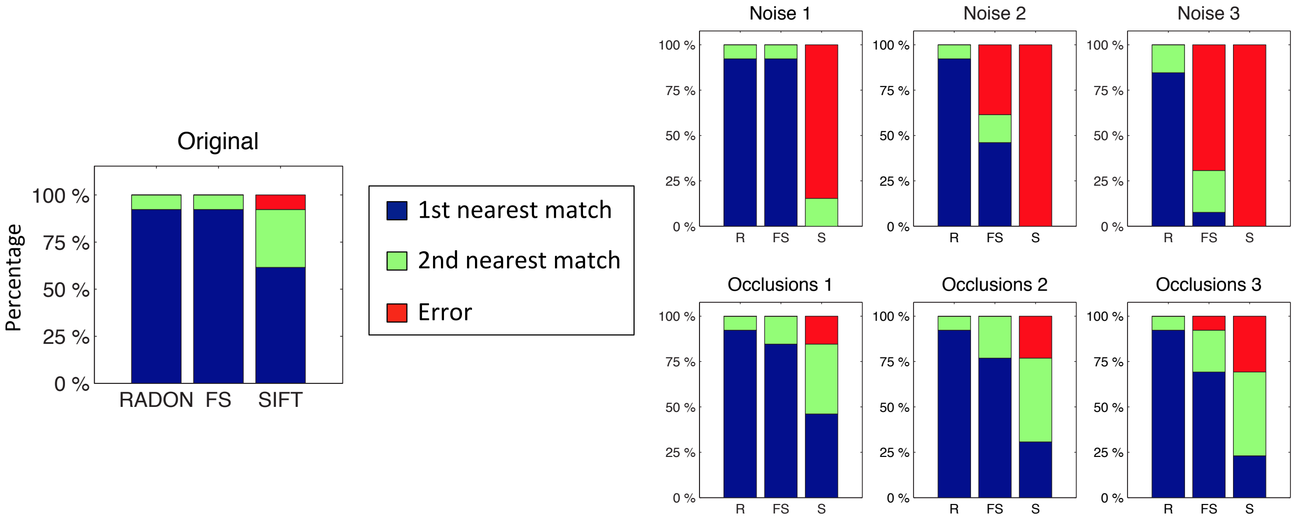
$$dist_{SIFT}(im_1, im_2) = \frac{1}{\sum_{i=1}^{n} \frac{1}{d_i}} \tag{16}$$

where d_i is the average distance between each SIFT feature in the first image and its corresponding feature in the second image. n is the number of matched SIFT features.

Figure 22a shows the success rate of the comparative experiment using the original test images. Thirteen random test images have been used to obtain this figure. Later, we repeated the experiment taking into consideration the presence of different levels of noise and occlusions on the test images, to test the robustness of each method under these typical situations. The results are shown in Figure 22b. Noises 1, 2 and 3 are random noises with maximum value equal to 20%, 40% and 60% of the maximum intensity. Occlusions 1, 2 and 3 imply that the test image has 10%, 20% and 30% occlusion, respectively.

As for the computational time used to do this experiment, the Radon transform spends 0.17 s; SIFT spends 12 s (using the algorithm of [6]). At last, the Fourier signature method needs 0.06 s.

Taking these results into account, the RT descriptor has been proven to be an effective method. It shows a reasonably good computational cost, and its robustness in the localization process is an important feature. While the FS and SIFT methods have worse behavior as noise and occlusions increase, RT has a good and stable behavior.



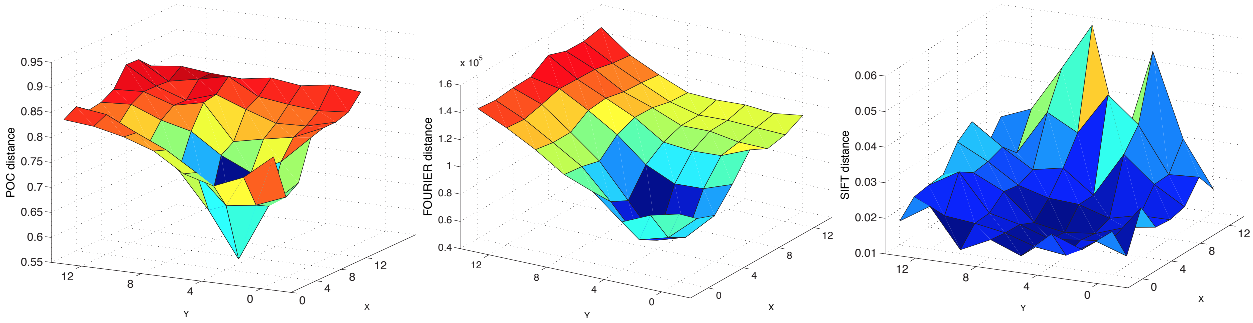
(**a**) (**b**)

Figure 22. Success rate with the laboratory database. (**a**) Using the original test images. (**b**) Taking into consideration the presence of different levels of noise and occlusions on the test images.

To explore the reason for this behavior, we depict in Figure 23 the distance between one sample test image and each image of the map. The test image is in the position $(X,Y) = (3,3)$. In this figure, the Radon descriptor stands out for its ability to localize the robot sharply since the distance increases quickly around the minimum. To check this performance, we have defined a parameter that measures the sharpness level of the minimum in each case (Equation (17)). The sharper the function, the higher is C.

$$C = \frac{mean(distance) - min(distance)}{max(distance) - min(distance)} \cdot 100(\%) \tag{17}$$

Using this equation with each descriptor, SIFT, FS and RT and carrying out 13 experiments in each case, the average value of C is: 60% using SIFT, 61% using FS and 79% using RT.



(a) (b) (c)

Figure 23. Distance between a sample test image captured at x = 3, y = 3 and all of the map images using (**a**) the Radon descriptor and POC distance, (**b**) Fourier signature (FS) and (**c**) SIFT.

7. Conclusions

In this paper we have presented two different methods to estimate the position of a robot in an environment in two different situations. In one of them, the robot knows the map positions, and in the other, it only knows the map images. The first method calculates the nearest position of the map with the test image taken by the robot, and the second method creates a local map with the nearest map images and the test image taken by the robot. These methods use omnidirectional images and global appearance descriptors.

We have also developed a method to build global appearance descriptors from omnidirectional images, based on Radon transform.

Both methods are first tested with our virtual database to determinate the best parameters, and they are then tested with actual databases. One of the methods has been analyzed with a database with changes in lighting conditions and adding objects. The other method has been analyzed with two different databases captured at our University.

Furthermore, a comparative analysis between the proposed descriptor and two other classical descriptors has been carried out. This analysis has shown the robustness of our method to solve the localization problem when the test images present noise or occlusions. These are common phenomena in real applications, and the RT method has been proven to be able to cope with them.

The results presented in this paper show the effectiveness of global appearance descriptors of omnidirectional images to localize the robot and to create a map. We are now working to improve these methods, and we are trying to carry out the localization in a map with more degrees of freedom. We are also working to include the color information of the images in the global appearance descriptors.

Acknowledgments

This work has been supported by the Spanish government through the project DPI 2013-41557-P: "*Navegación de Robots en Entornos Dinámicos Mediante Mapas Compactos con Información Visual de Apariencia Global*" and by the Generalitat Valenciana (GVa) through the project GV /2015/031: "*Creación de mapas topológicos a partir de la apariencia global de un conjunto de escenas*".

Author Contributions

The work presented in this paper is a collaborative development by all of the authors. Yerai Berenguer and Oscar Reinoso defined the research line. Mónica Ballesta and Yerai Berenguer carried out the preliminary experiments with descriptors and distances to tune the parameters. Yerai Berenguer, Luis Payá and Oscar Reinoso designed and implemented the mapping and localization algorithms. At last, Luis Payá, Yerai Berenguer and Mónica Ballesta carried out the mapping and localization experiments with the robotic platform.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Winters, N.; Gaspar, J.; Lacey, G.; Santos-Victor, J. Omni-directional vision for robot navigation. In Proceedings of the IEEE Workshop on Omnidirectional Vision, Hilton Head Island, SC, USA, 12 June 2000; pp. 21–28.
2. Valiente, D.; Gil, A.; Fernández, L.; Reinoso, O. A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images. *Robot. Auton. Syst.* **2014**, *62*, 108–119.
3. Maohai, L.; Han, W.; Lining, S.; Zesu, C. Robust omnidirectional mobile robot topological navigation system using omnidirectional vision. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1942–1952.
4. Garcia-Fidalgo, E.; Ortiz, A. Vision-based topological mapping and localization by means of local invariant features and map refinement. *Robotica* **2015**, *33*, 1446–1470.
5. Garcia-Fidalgo, E.; Ortiz, A. Vision-based topological mapping and localization methods: A survey. *Robot. Auton. Syst.* **2015**, *64*, 1–20.
6. Lowe, D. Object Recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision, Corfu, Greece, 20–27 September 1999; pp. 1150–1157.
7. Bay, H.; Tuytelaars, T.; Gool, L. SURF: Speeded up robust features. *Comput. Vis. ECCV* **2006**, *3951*, 404–417.
8. Chang, C.; Siagian, C.; Itti, L. Mobile robot vision navigation and localization using GIST and saliency. In Proceedings of the International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4147–4154.
9. Payá, L.; Fernández, L.; Gil, L.; Reinoso, O. Map building and Monte Carlo localization using global appearance of omnidirectional images. *Sensors* **2010**, *10*, 11468–11497.
10. Wu, J.; Zhang, H.; Guan, Y. An efficient visual loop closure detection method in a map of 20 million key locations. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 861–866.
11. Payá, L.; Amorós, F.; Fernández, L.; Reinoso, O. Performance of Global-Appearance Descriptors in Map Building and Localization Using Omnidirectional Vision. *Sensors* **2014**, *14*, 3033–3064.
12. Radon, J. Uber die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten. *Ber. Verh. Sacks. Akad.* **1917**, *69*, 262–277.
13. Oliva, A.; Torralba, A. Building the gist of a scene: the role of global image features in recognition. In *Visual Perception Fundamentals of Awareness: Multi-Sensory Integration and High-Order Perception*; Elsevier: Amsterdam, The Netherlands, 2006; pp. 23–36.
14. Hoang, T.; Tabbone, S. A Geometric Invariant Shape Descriptor Based on the Radon, Fourier, and Mellin Transforms. In Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2085–2088.
15. Hasegawa, M.; Tabbone, S. A Shape Descriptor Combining Logarithmic-Scale Histogram of Radon Transform and Phase-Only Correlation Function. In Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR), Beijing, China, 18–21 September 2011; pp. 182–186.

16. Torralba, A.; Murphy, K.; Freeman, W.; Rubin, M. Context-based vision system for place and object recognition. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 273–280.
17. Siagian, C.; Itti, L. Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 300–312.
18. Oppenheim, A.; Lim, J. The importance of phase in signals. *IEEE Proc.* **1981**, *69*, 529–541.
19. Kuglin, C.; Hines, D. The phase correlation image alignment method. In Proceedings of the IEEE International Conference on Cybernetics and Society, San Francisco, CA, USA, 23–25 September 1975; pp. 163–165.
20. Kobayashi, K.; Aoki, T.; Ito, K.; Nakajima, H.; Higuchi, T. A fingerprint matching algorithm using phase-only correlation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2004**, *87*, 682–691.
21. Seber, G.A.F. *Multivariate observations*; John Wiley & Sons, Inc: New York, NY, USA, 1984.
22. Ishiguro, H.; Tsuji, S. Image-based memory of environment. In Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems' 96 (IROS 96), Osaka, Japan, 4–8 November 1996; pp. 634–639.
23. Menegatti, E.; Maeda, T.; Ishiguro, H. Image-based memory for robot navigation using properties of omnidirectional images. *Robot. Auton. Syst.* **2004**, *47*, 251–267.
24. Vardy, A.; Möller, R. Biologically plausible visual homing methods based on optical flow techniques. *Connect. Sci.* **2005**, *17*, 47–89.