

Research Article

Global Appearance Applied to Visual Map Building and Path Estimation Using Multiscale Analysis

**Francisco Amorós,¹ Luis Payá,¹ Oscar Reinoso,¹
Walterio Mayol-Cuevas,² and Andrew Calway²**

¹ *Department of Systems Engineering and Automation, Miguel Hernández University, Avenida de la Universidad s/n, 03202 Elche, Spain*

² *Department of Computer Science, University of Bristol, Woodland Road, Bristol BS8 1UB, UK*

Correspondence should be addressed to Francisco Amorós; famoros@umh.es

Received 7 March 2014; Revised 11 June 2014; Accepted 11 June 2014; Published 22 July 2014

Academic Editor: Yi Chen

Copyright © 2014 Francisco Amorós et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work we present a topological map building and localization system for mobile robots based on global appearance of visual information. We include a comparison and analysis of global-appearance techniques applied to wide-angle scenes in retrieval tasks. Next, we define multiscale analysis, which permits improving the association between images and extracting topological distances. Then, a topological map-building algorithm is proposed. At first, the algorithm has information only of some isolated positions of the navigation area in the form of nodes. Each node is composed of a collection of images that covers the complete field of view from a certain position. The algorithm solves the node retrieval and estimates their spatial arrangement. With these aims, it uses the visual information captured along some routes that cover the navigation area. As a result, the algorithm builds a graph that reflects the distribution and adjacency relations between nodes (map). After the map building, we also propose a route path estimation system. This algorithm takes advantage of the multiscale analysis. The accuracy in the pose estimation is not reduced to the nodes locations but also to intermediate positions between them. The algorithms have been tested using two different databases captured in real indoor environments under dynamic conditions.

1. Introduction

The autonomous navigation of a mobile robot usually involves a minimal knowledge of the surrounding environment. Normally, that knowledge is used with the purpose of building an internal representation of the area in a map. Using the map and the current information that the robot receives from its sensors, it is possible to carry out the localization of the robot and also to simultaneously add new information to the map.

In the literature, we can find a wide variety of environment representations depending on the sensor used. In this way, it is possible to find examples that try to compute the position of the robot using GPS, laser, or wheel encoders as input information sensors. Among all the possibilities, vision systems have become common sensors for robot control due to the richness of the information they provide, their

relative low weight and cost, and the variety of possible configurations. So that, we can find researches based on single standard cameras as [1], wide-angle cameras [2], stereo cameras [3], catadioptric systems that provide us with omnidirectional images [4], or an array of cameras arranged circularly to obtain a panoramic image [5]. In this work, we use a fish-eye single camera, due to the fact that they provide a wide-angle view of the environment and have lower cost than other visual systems.

In the great majority of real visual applications, it is not possible to work with the image information directly from the sensors, as the memory requirements and computational cost would make the process unfeasible. Taking this into account, it is necessary to find an alternative representation of images that contains as much information as possible with a reduced memory size. In this task, two main categories can be found: feature based and global-appearance based

descriptors. The first approach is based on the extraction and description of significant points or regions from the scene. In this sense, we find examples of the use of SIFT features [6, 7], SURF [8, 9], or Harris edge and corner detector [10] applied to localization and mapping tasks. On the other hand, global-appearance descriptors try to describe the scene as a whole, without the extraction of local features or regions. These techniques have a special interest in unstructured and changing environments where finding patterns to recognize the scene might be difficult. For example, Kröse et al. [11] demonstrate the robustness of PCA (principal component analysis) applied to image processing. Menegatti et al. [12] take advantage of the properties of the discrete fourier transform (DFT) applied to panoramic images in order to build descriptors of the scene. In [13], Kunttu et al. describe the behaviour of a descriptor based on Fourier transform and Wavelet filter in image retrieving tasks.

Regarding the representation of the map, three main approaches stand out: metric, topological, and hybrid techniques. Metric maps include information of distances with respect to a predefined coordinate system. These maps provide the position of the robot except for an uncertainty associated with the sensor error. However, they usually have high computational cost. As an example, it is possible to find examples based on a sonar sensor applied to robot navigation [14] and other approaches that solve the SLAM problem using a team of robots with a map represented by the three-dimensional position of visual landmarks [15].

In contrast, topological techniques use graph-based representations of the environment. The nodes correspond to different areas of the environment, whereas the edges represent the connectivity relationships between the nodes. In those maps there are no absolute distances. Since they constitute a simple and compact representation, time and memory requirements are generally lower than in metric maps. However, they contain enough information to allow the robot to navigate autonomously through the environment. Many different visual-based navigation systems can be found, as [16, 17] present. They both use an omnidirectional camera as input sensor and topological maps as a representation of structured indoor office environments. In these maps, the nodes correspond to images of the areas where the robot navigates. These images are described using PCA techniques in order to reduce the memory requirements. The navigation between nodes relies on a visual path following algorithm that extracts the edges of corridor walls. A similar system but using a single camera is developed in [18]. Nodes correspond to special locations where certain actions must be taken, such as a turning or a door crossing; meanwhile, the edges represent trajectories where visual servoing navigation can be carried out. Specifically, the visual servoing is based on the vanishing point to keep the trajectory of the robot centered in the corridor. Other examples of topological map indoor navigation can be found in [19], where the gradient orientation histogram of the image is used in order to describe the scenes. Štimec et al. [20] present an appearance-based method for path-based map learning in both indoor and outdoor environments. This system is based on clustering

of PCA features extracted from panoramic images in order to obtain distinctive visual aspects. In [21] we can find a topological localization system in a home environment. They use a sonar sensor and a grid map matching in order to carry out the localization of a robot, dealing also with the kidnapping problem. In [22], another example of topological homing navigation system is presented. In the proposal, some information from an omnidirectional visual system, a 3D stereo vision system, and the odometry are combined to carry out mapping and localization tasks using a mobile robot. FAB-MAP [23, 24] is another well-known topological SLAM approach, based on SURF features extraction to describe the appearance of the images. This algorithm has been tested in large scale navigation environments. On the other hand, Bellotto et al. [25] describe a visual topological localization system for mobile robot that uses digital image zooming. This work is based on the appearance of omnidirectional images and includes an image matching algorithm that improves the image association by means of the use of digital zoom of the scenes.

At last, regarding hybrid techniques, they try to take advantage of both topological and metric proposals. Normally, hybrid maps use metric in order to build local maps of separated areas, whereas topological relations are used in order to create a general map. It is also possible to introduce the topological relations to carry out loop closures in metric maps. An example of hybrid SLAM algorithm is RatSLAM [26]. This technique integrates the internal odometry of the robot provided by wheel encoders and visual information, using a neural network in a biologically inspired fashion. In [27] we can find the joint use of FAB-MAP and RatSLAM.

In this paper we propose a framework for only-visual topological map building and localization. Our technique stands out because of the use of global-appearance techniques to describe the scenes and the application of a multiscale analysis of the visual information to estimate relative distances between images. The system is intended for autonomous robotic navigation in mainly indoor spaces, such as offices and industrial environments where topological navigation is suitable.

The map is represented as a graph. In this graph, the nodes are collections of 8 wide-angle images captured every 45 degrees, covering the complete field of view around their positions. The topological distance between nodes, which is estimated by means of the multiscale analysis, will be proportional to the actual distance between positions of the nodes in the real world.

We use the information of several routes of images acquired along the environment, which pass through the nodes, to carry out the map building. The map building system is able to recognize new nodes, find their orientation, their relative position, and connectivity using these routes of images. We use the multiscale analysis to obtain both an increase of correct matching of route images in the map database and also a measurement of the relative position of the compared scenes.

After the map building, we propose a route estimation algorithm, which takes also advantage of that multiscale analysis. In this case, this analysis is used to enhance

the localization of the robot, being able to locate the robot not only in the nodes positions but also in intermediate points. We also introduce a weighting function that improves the localization accuracy using the graph obtained during the map building.

In [28], we find an example of visual route navigation using visual information that tries to keep the input memory to a minimum. Following that idea, we include a study of the computational cost of image retrieving using different global-appearance descriptors and image sizes in order to minimize the time and memory requirements. This study will be used to select the descriptor and the features of the images in the map building and localization experiments.

The following glossary includes some of the terms used in the text.

- (i) *Node*: collection of 8 images captured from the same position on the ground plane every 45° approximately, covering the complete field of view around that position.
- (ii) *Map database or node database*: collection of the images of all the nodes.
- (iii) *Map*: graph that represents the topological layout of the nodes.
- (iv) *Map building*: process of finding the topological connection between nodes and their relative position.
- (v) *Topological distance*: relative position between images or nodes in the map.
- (vi) *Image distance (d)*: Euclidean distance between the descriptors of two images.

The contributions of this paper are the following.

- (i) We propose the multiscale analysis, which allows us to determine the relative position of two images captured with approximately the same orientation using global-appearance descriptors.
- (ii) The paper includes a map building algorithm. Our algorithm lies in finding the spatial distribution of a collection of nodes of images distributed along an area. The system provides a topological graph that represents the adjacency relations and layout of the nodes.
- (iii) We develop a topological localization system that extends the possible current pose estimation of the robot not only to the node locations but also to intermediate positions, making use of the multiscale analysis. This system allows the robot to estimate its path during the navigation using only the visual information.
- (iv) Finally, we offer an experimental validation of the map building and route path estimation algorithms using our own database, which contains information of two different areas.

The remainder of the work is structured as follows. Section 2 introduces the specifications of the images captured

to carry out the experiments and the global appearance descriptors considered to describe the scenes. In Section 3 we compare and analyze the computational requirements and matching precision of different descriptors using several image resolutions. Section 4 introduces the multiscale analysis. Next section presents the algorithm developed to build the topological map. In Section 6, we explain a novel route path estimation algorithm. Section 7 includes the experimental results using our own database. Finally, a summary with the main contributions in this work is included.

2. Images Features and Descriptors

This section describes the main features of the images used in the experimental part and the techniques we have applied in order to obtain a descriptor that extracts the main information from the images based on their global appearance.

The images are captured using a fisheye lens camera. Specifically, we use the Hero2 camera of GoPro [29]. We choose this camera due to its wide-angle field of view (127°), its low weight, and relative low cost compared to other visual systems.

The goal of the image descriptors is to solve the problem of place recognition using the global appearance of the scenes, trying to keep the memory requirements and computational cost to a minimum. The descriptors based on the global appearance concentrate the visual information of the image working with it as a whole, avoiding the extraction of landmarks or local features. They have presented good results in visual navigation tasks. It is possible to find previous works comparing these techniques [30, 31] or using them in map building and localization [32]. These researches use omnidirectional vision sensors. However, we do not have knowledge of any work where these techniques had been applied to nonpanoramic images.

Due to the use of a fisheye lens, the images captured with our camera present a radial distortion that must be corrected. It would be impossible to obtain useful information from the distorted images using global appearance descriptors, since they are based on the spatial distribution and disposition of the elements in the scene. For that reason, we use the Matlab Toolbox *OCamCalib* [33] to calibrate the camera and compute the undistorted scenes from the original images.

We consider UV the coordinate system of the image and XYZ the coordinate system of the real world, which is situated in the focal point of the lens. We consider the UV directions aligned with XY . The coordinates x and y of a point P in the real world are proportional to the coordinates u and v of an image point:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \cdot \begin{bmatrix} u \\ v \end{bmatrix}. \quad (1)$$

Therefore, the vector \vec{P} can be defined as

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \cdot u \\ \alpha \cdot v \\ f(u, v) \end{bmatrix}. \quad (2)$$

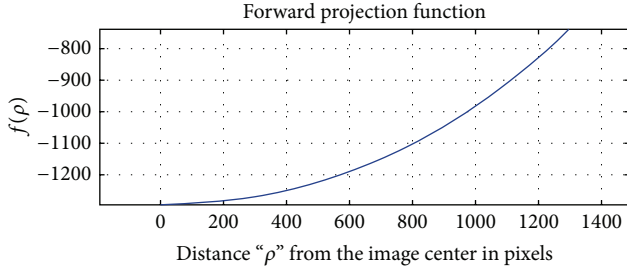


FIGURE 1: Forward projection function of the camera.

We can include the parameter α in the function $f(u, v)$. In this way, the previous equation can be expressed as

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix}. \quad (3)$$

Due to the symmetric geometry of the lens, the coordinate z of the point P only depends on the distance of the image point p regarding the image coordinate center:

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix}, \quad \text{with } \rho = \sqrt{u^2 + v^2}. \quad (4)$$

The function f , also named forward projection function, depends on the lens geometry. In general, it can be expressed as a n -degree polynomial:

$$f(\rho) = a_0 + a_1 \cdot \rho + a_2 \cdot \rho^2 + a_3 \cdot \rho^3 + \dots + a_n \cdot \rho^n. \quad (5)$$

In our particular case, the minimum calibration error is obtained for a polynomial of degree equal to 3. The calibration function is

$$f(\rho) = -13194.89 + 2.252 \cdot 10^{-4} \rho^2 + 5.90 \cdot 10^{-8} \rho^3. \quad (6)$$

This function provides information about the direction of the rays that arrive to the camera system. The undistorted image corresponds to the projection of these rays in a plane parallel to the camera sensor.

Figure 1 shows the forward projection function of the camera, obtained after the calibration process.

In Figure 2 we can see an example of the original image and the undistorted view.

In the remainder of the paper, when we talk about the images, we will refer to the undistorted version of the original scenes.

Next, we include a summary of different descriptors based on the global appearance of the scenes.

2.1. Fourier-Based Techniques. It is possible to describe an image using the discrete Fourier transform over its rows. We can transform each row of the image $\{a_n\} = \{a_0, a_1, \dots, a_{N-1}\}$ into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \dots, A_{N-1}\}$:

$$\{A_n\} = \mathcal{F}[\{a_n\}] = \sum_{n=0}^{N-1} a_n e^{-j(2\pi/N)kn}; \quad k = 0, \dots, N-1. \quad (7)$$

The most relevant information of the image is concentrated in the low frequencies. These frequencies represent large scale features of the images. Moreover, in real images, high frequencies are often affected by noise. Figure 3 shows the modules of the first components of the Fourier transform of each row of an image. Hence, we select only the first coefficients of the discrete Fourier transform of each row to build the descriptor.

In [12], Menegatti et al. present a descriptor that uses the discrete Fourier transform in panoramic images, defining the Fourier signature. Since the magnitude of the transform presents rotational invariance, in that case it constitutes the localization descriptor. However, our database images are not panoramic, and the rotational invariance may introduce localization errors in areas where there is a symmetry between different images, as corridors. To avoid it, our descriptor is not made up by the magnitude but by the original complex values.

2.2. Histogram of Oriented Gradients. The descriptor based on the histogram of oriented gradients (HOG) [34] uses the orientation of the gradient of an image. First we have to compute the spatial derivatives of the image along x and y -axis (I_x and I_y). Then, we obtain the magnitude and direction values of the gradient of each pixel:

$$|G| = \sqrt{I_x^2 + I_y^2}; \quad \theta = \arctan\left(\frac{I_y}{I_x}\right). \quad (8)$$

Next, the image is divided into cells, and the histograms of the cells are computed. In Figure 4 we can see the division of the gradient of an image to obtain different cells (Figure 4(b)) and the estimation of the histogram of each cell (Figure 4(c)). The histogram is computed based on the gradient orientation of the pixels within the cell, weighted with the magnitudes of the gradient. The descriptor consists of the histograms' values of all the cell the image is divided into, ordered in a vector.

2.3. Gist-Based Techniques. Gist denotes a group of techniques that can be used to compress visual information as [35] details. These descriptors try to obtain the essential information of the images simulating the human perception system, that is, identifying a scene through its colour or remarkable structures, avoiding the representation of specific objects. Oliva and Torralba [36] develop this idea under the name of *holistic representation of the spatial envelope* to create a descriptor. In [37], this model uses global scene features, such as spatial frequencies and different scales based on Gabor filtering.

A Gabor filter is a lineal filter whose impulse response is a sinusoid modulated with a Gaussian function [38]. Therefore, a Gabor mask is localized both in the spatial and in the frequency domains (Figure 5). Thanks to its properties regarding textures treatment, Gabor filter can be used in compression and segmentation of digital images, as [39] shows.

First, we create a bank of Gabor masks including different resolutions and orientations. Then, the image is filtered using the set of filters. The orientation of each filtering depends



FIGURE 2: (a) Image captured with a fisheye lens camera and (b) its corresponding undistorted view.

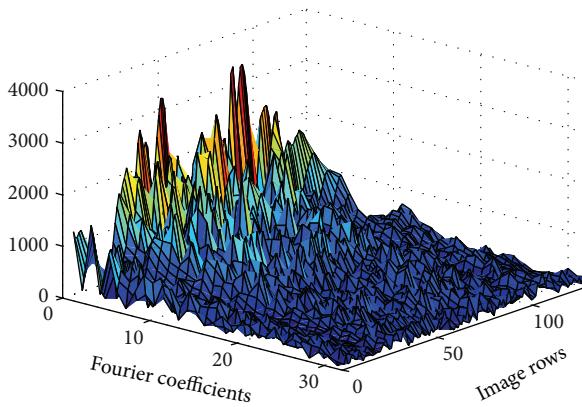


FIGURE 3: Module of the discrete Fourier transform of an image per rows.

on the number of masks of each resolution, since they are equally distributed between 0 and 180 degrees. The filtered images encode different structural information according to the mask applied. After that, the images are divided into cells, and we compute the average pixels' value within each cell. This process is represented in Figure 6. This is repeated for every filtered image. The final descriptor is composed of the mean value of intensity of the pixels contained in horizontal cells applied to every filtered scene.

3. Localization Recognition

In this section, we present a comparison between the different global appearance descriptors included in the previous section applied to location retrieval tasks. The aim of this study is to check the performance of these descriptors applied over perspective images using different resolutions. The comparison takes into account both the precision in correct matching and the computational requirements.

The database is composed of several nodes of images captured in different isolated places randomly chosen in both indoor and outdoor environments. Note that every node is composed of 8 images captured with a phase lag equal to 45 between consecutive images. We also take a set of test images.

TABLE 1: Image resolutions used in the experiments.

	Image's pixels
Size 1	1817×1004
Size 2	512×283
Size 3	256×128
Size 4	128×64
Size 5	64×32
Size 6	32×16
Size 7	16×8

The test images are captured in the same locations of the nodes with unknown orientations. Specifically, we capture 26 nodes and 3 test images per location. This database is different from the images used in the following sections.

In the experiments, we create a database with the descriptors of all the images of the nodes. When a new test image arrives, we compute its descriptor and compare it with the database. We define the image distance as the Euclidean distance between descriptors, which allows us to measure the similarity between scenes. Regarding the classifier, we choose the nearest neighbour.

Moreover, we are interested in finding the minimum image resolution we can use without detriment of matching precision. Table 1 shows the image sizes we have tested along the experiments. Size 1 is the original resolution of the camera.

In Figure 7 we can see the necessary time to compute the descriptor of an image. The information has been divided into two different graphs in order to clarify the data included. On the other hand, Figure 8 shows the memory requirements to store the image descriptors of all the nodes in a database. We can appreciate an exponential reduction of the requirements when we use smaller resolutions.

Gist-Gabor stands out as the computationally most expensive descriptor (Figure 7), although as the image size is reduced, the time differences between descriptors decrease. Regarding the memory requirements (Figure 8), the Fourier signature is the technique with higher memory requirements. Figure 9 shows the performance of each descriptor when

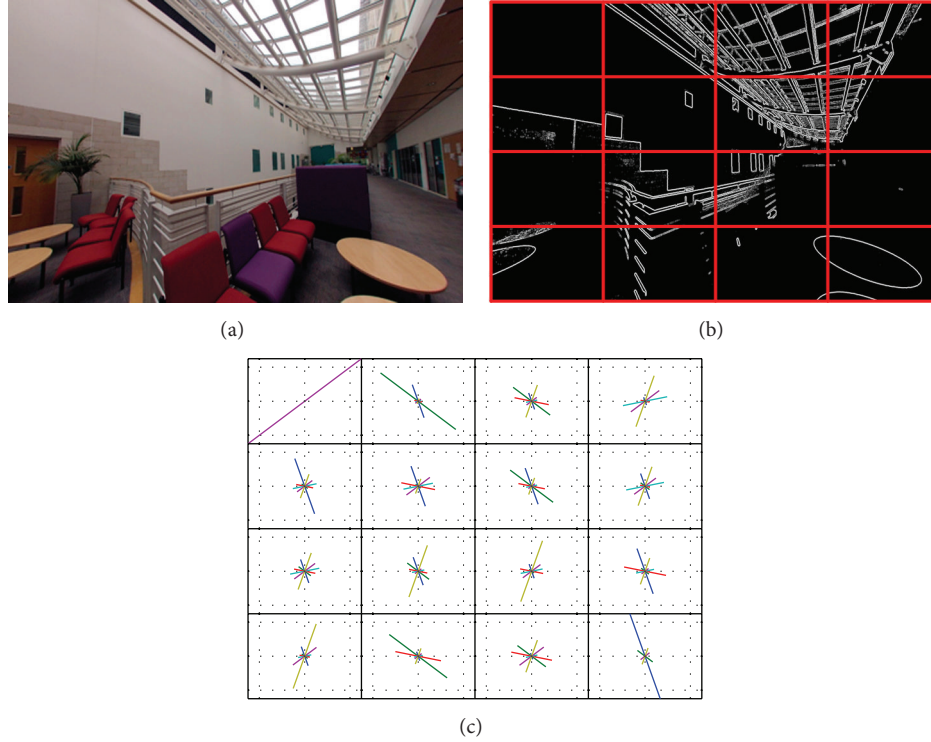


FIGURE 4: Example of HOG description process. (a) Original image, (b) gradient of the image and cell division, and (c) histogram of oriented gradients of each cell.

finding the correct position using recall and precision measurements [40] for the three nearest neighbours. The Fourier signature is almost invariant to the image resolution, whereas HOG presents a notable reduction of the precision using the smallest resolutions.

The main criteria for the selection of the descriptor is the precision in image association. For that reason, HOG becomes the less appropriate descriptor, although its memory and time performance are favorable. The Fourier signature presents a high precision in the position estimation for all image sizes, but it is lower than Gist-Gabor. Moreover, the size of the database created when we use the Fourier signature is higher than the other techniques. For those reasons, the descriptor selected is Gist-Gabor. The results obtained using the fifth image size (64×32 pixels) show an appropriate compromise between time and memory requirements and precision.

4. Multiscale Analysis

This section describes the multiscale analysis. During the matching process between the images of the nodes and the routes, the nodes might be too separated for a correct association, especially in the route locations that are halfway between nodes. As a consequence, the appearance of the route images could present insufficient similarity with the nodes scenes to find a reliable retrieval in the node database. The aim of the multiscale analysis is to improve the association

accuracy and to estimate the relative position between images making use of the global appearance descriptors.

Given two images, this technique carries out the comparison of several zooms-in of the central area of each image at different scales. Figure 10 shows the field of view of a camera when it moves forward perpendicularly to its projection plane. We can appreciate that the scene in the ahead position, represented in blue, corresponds to the central area of the field of view associated with the first position, represented in orange. If we select the central area of the orange image and rescale it to the original image size (simulating a digital zoom), the appearance regarding the second image (the blue) increases. Figure 11 illustrates an example of this idea. It includes two images captured during the forward navigation movement of robot following a route (Figures 11(a) and 11(b)). In the figure is also included a zoom-in of the central area of (a) (Figure 11(a')). We can appreciate that the zoomed image (a') is more similar in appearance to (b) than the original scene (a).

The similarity between scenes is measured using global appearance descriptors (Section 3). After the comparisons with different scales, we select the association with the lowest image distance (i.e., the nearest neighbour), since it denotes the most similar images using the global appearance.

The scales of the two images matched during the association process provide information about their relative position. Specifically, the algorithm uses the difference of the scales to estimate the distance between images.

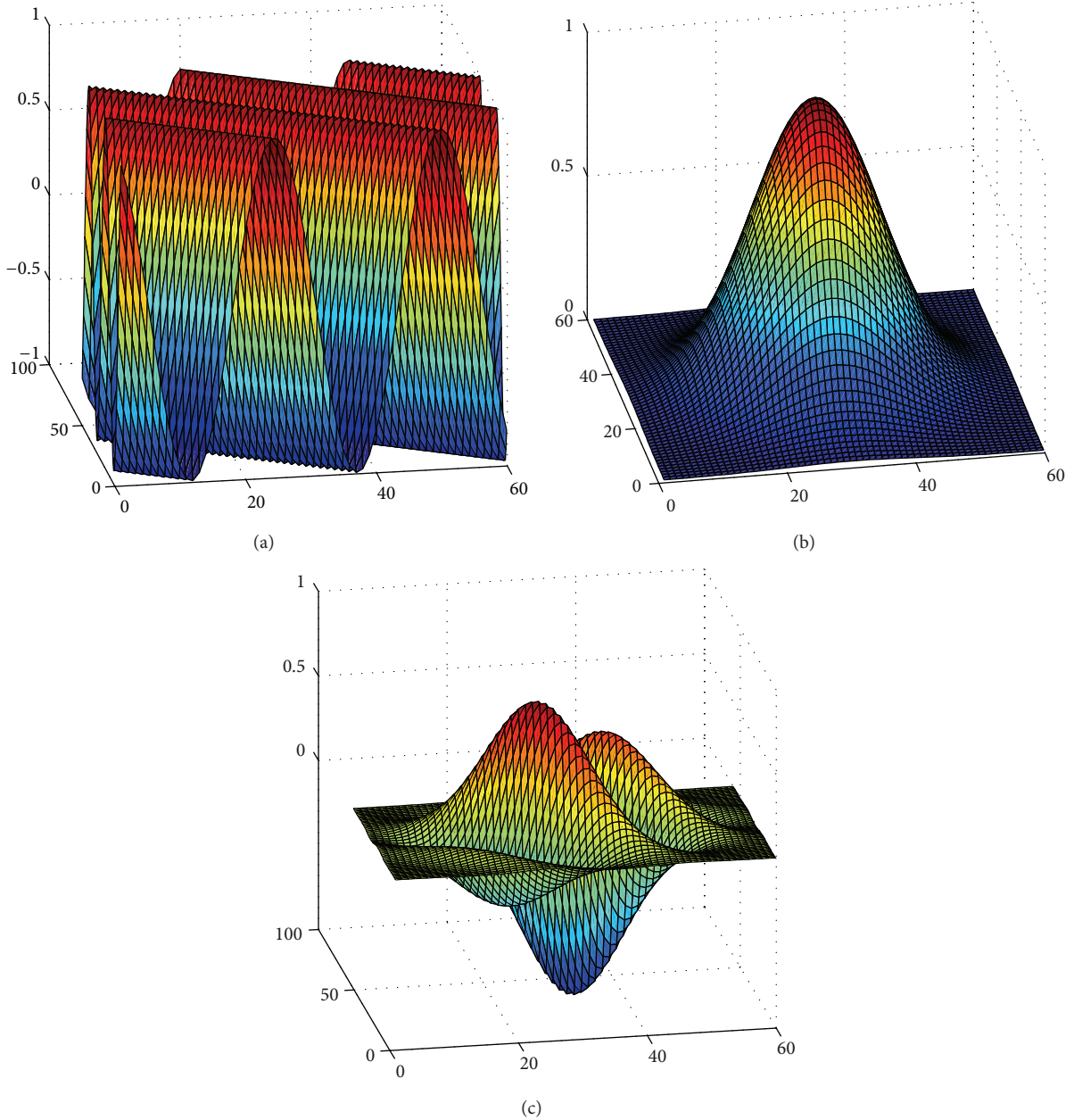


FIGURE 5: (a) Complex sinusoid, (b) Gaussian envelope, and (c) Gabor filter resulting of the convolution of both functions.

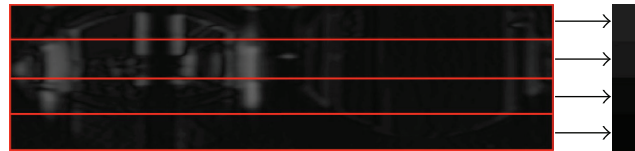


FIGURE 6: Division of the filtered image in cells and estimation of the mean intensity value of the pixels within each cell.

Figure 12 illustrates this process. The example includes four images of a route captured sequentially as the camera moves forward. In the example, we aim to estimate the topological distance of the four scenes regarding Scene 1, which is our reference image. For that purpose, we estimate

different scales of Scene 1, compute their global-appearance descriptors, and compare them with (a) Scene 1, (b) Scene 2, (c) Scene 3, and (d) Scene 4 without zoom.

Since these images are captured sequentially, each scene is geometrically more separated from Scene 1 in the real world.

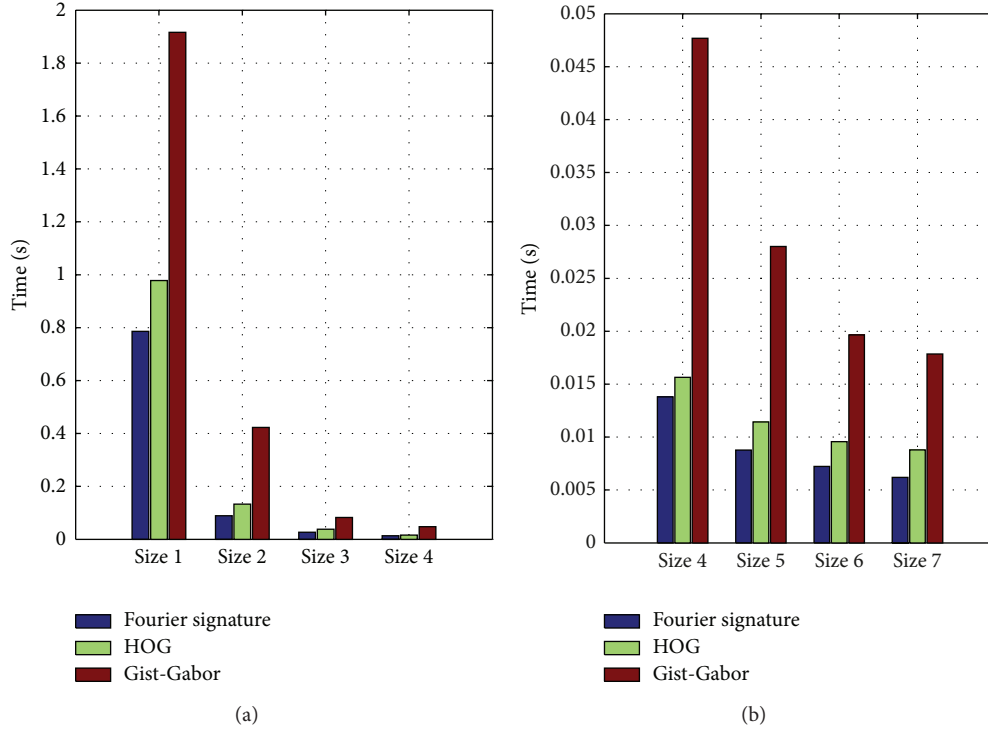


FIGURE 7: Necessary time to build each descriptor depending on the image size.

In other words, Scene 3 is more separated from Scene 1 than Scene 2, and Scene 4 has been captured in the most distant point from Scene 1.

On the other hand, Figure 12 includes on the right side a graph that represents the image distances of the four scenes versus the different scales of Scene 1.

The scale factor (s) is the quotient between the original resolution of the image and the size of the area we select. For instance, if the resolution of the image is 32×64 , a scale equal to $s = 2$ is supposed to select the 16×32 central pixels.

In the graph, we highlight the scale of Scene 1 that presents the minimum image distance for each scene of the example. Note that lower image distances (i.e., Euclidean distance between descriptor) denote higher similarity between scenes.

As expected, the minimum image distance comparing with Scene 1, which is the same image than the reference scene, is obtained using a scale equal to 1, that is, when no zoom is applied. Regarding the comparisons with the other scenes, we can realize that the minimum image distance is obtained for higher zoom scales as the scene is more separated geometrically in the real world from the reference image (Scene 1).

Therefore, there is a direct correlation between the scale where the minimum image distance is obtained and the geometrical distance of the scenes in the real world. In our map building and localization algorithms, we use the difference of scales as topological distance between scenes.

Moreover, as seen in Figure 12, we obtain also a reduction of the image distance when comparing two images, which

means that we increase the similarity of the compared scenes using the global appearance.

This increase in the similarity between images turns into an improvement of the precision in the image association task. The map building and navigation algorithms proposed in the following sections of this work rely on the matching between the images of isolated positions in the environment (the nodes) and images acquired along routes. For that reason, an improvement of the association precision is important.

To measure this improvement, we study the association between 352 node images and 172 images of routes. We compare each image of the route with all the node images and select the association with minimum image distance (the nearest neighbour). We consider that the association has been correct when the selected node is the nearest in geometric distance in the real world. Figure 13 shows the recall-precision results using the multiscale analysis and without it. Thanks to the introduction of the multiscale analysis, the precision in correct node retrieval increases 14%.

Therefore, the multiscale analysis improves the association between images and provides a measurement of topological displacement between two images by means of the zoom scales (s).

5. Map Building

This section details our topological map building algorithm. It starts with a database that contains the descriptors of all the images of the nodes, with no information of their spatial

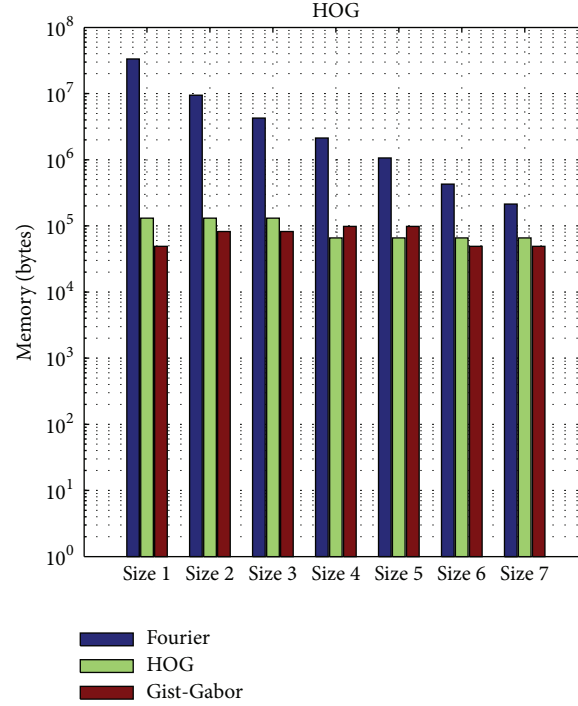


FIGURE 8: Database memory requirements using several image sizes and descriptors.

distribution. In the database, the images of the same node are stored consecutively, but the order of the nodes does not provide information about their spatial layout.

We also have different routes of images captured while the robot navigates along the nodes. The scenes of the routes are ordered as they are captured during the navigation, and the algorithm incorporates their information sequentially, that is, in the same order they are captured.

The aim of this algorithm is to select the nodes of the database as they are associated with the images of the routes using the global appearance descriptors, to establish the adjacency relationship between nodes, to define its orientation, and to estimate the distances between nodes using the multiscale analysis.

After the map building process, we obtain a graph that represents the spatial distribution of the nodes, and the edges are the adjacency relations between those nodes.

5.1. Estimation of the Relative Position between Nodes and Route Images Using the Multiscale Analysis. During the map building, the algorithm uses both the multiscale analysis to compare each image of the route with the images of the nodes. Given a route image, the matching process carries out the comparison of different scales of that image with several scales of the node images. After the comparisons, we select the experiment with the minimum image distance. s_n and s_r represent the specific scale factors of the node and the route images, respectively, obtained with the multiscale analysis.

These two scales permit determining the relative position between the image of the node and the route. The topological

distance (l) between the route image and the node can be estimated as

$$l = s_n - s_r. \quad (9)$$

Following the example included in Figure 14, when the route image is in front of the node (example Node 1), the comparison with the highest similarity between scenes is obtained doing a zoom-in of the node image (Figure 14(a')). Hence, $s_n > s_r$, obtaining a negative topological distance ($l > 0$). On the contrary, in the image of the route is situated backwards the node (example Node 2), the minimum image distance is obtained when we compare the image of the route using zoom-in with the image of the node without any zoom. In that case, according to (9), we obtain $l < 0$. Therefore, the topological distance l not only provides information about the relative distance between the nodes and the route images but also the direction of their distance by means of its sign.

In Figure 15, an example of a reduced experiment of node retrieval and distance estimation is shown. It includes two node images and nine images of a route whose path coincides with the nodes position. The results show the nearest node n , the scales of the node image (s_n) and the route image (s_r) estimated using the multiscale analysis, and the topological distance l . In this example, we have omitted the estimation of the orientation, since the route follows a straight line. In the localization results, we can see that the topological distance l is negative when the route images are backwards the nearest node and positive when they are ahead the node.

5.2. Association between Routes and Nodes Images. The first step in the map building algorithm is the matching between

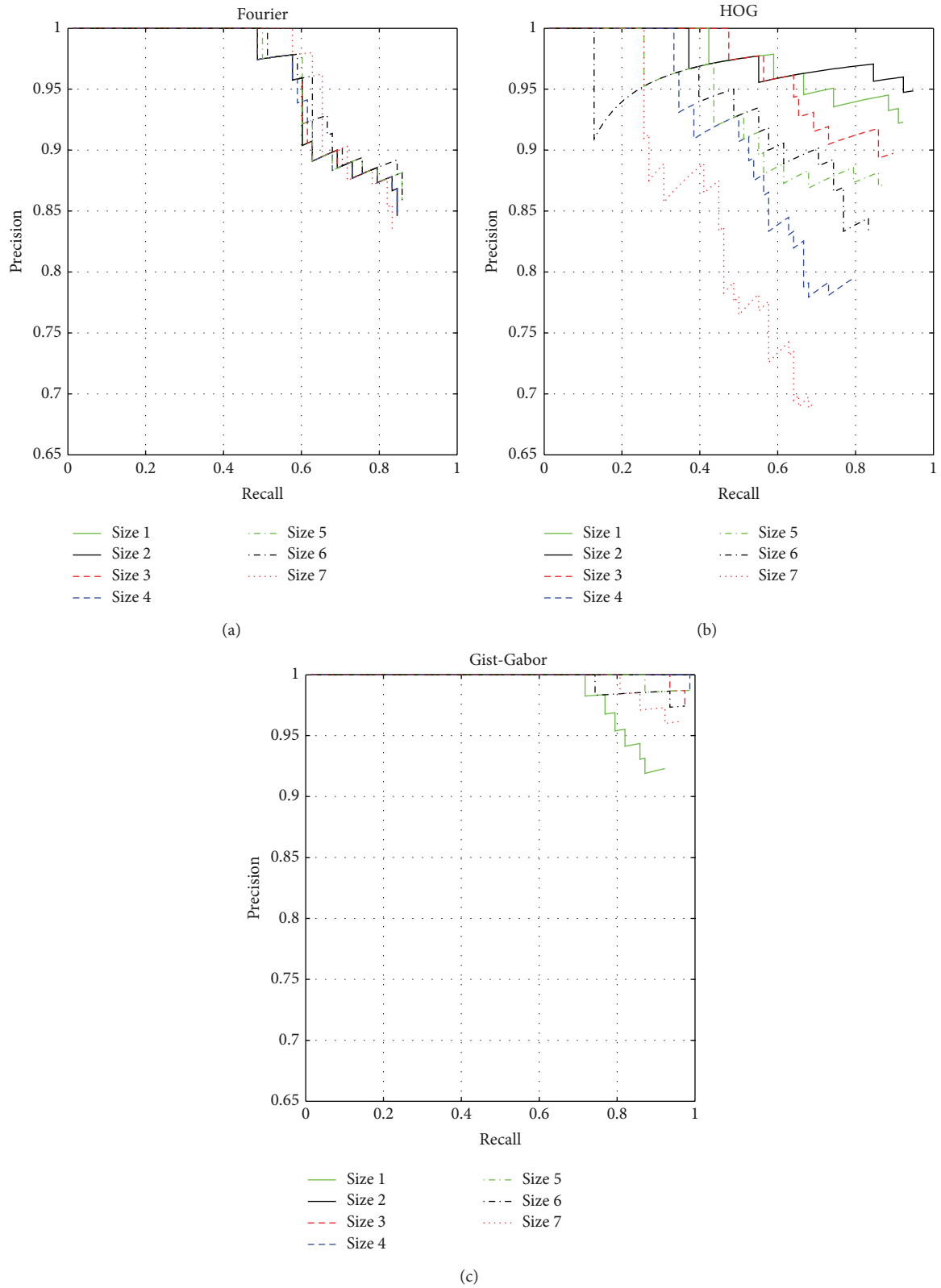


FIGURE 9: Recall-precision graphs in location retrieval considering the three nearest neighbours for different image's sizes using (a) Fourier signature, (b) HOG, and (c) Gist-Gabor.

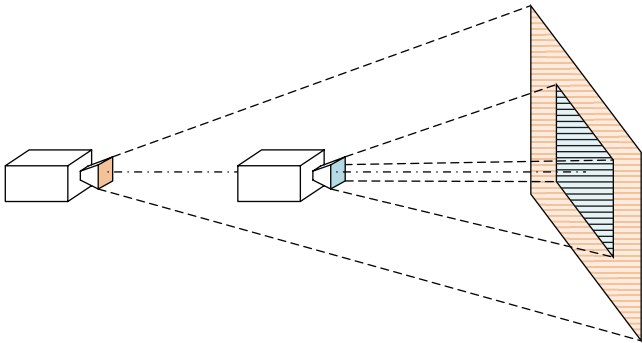


FIGURE 10: Representation of the field of view of a camera considering a movement perpendicular to the projection plane.

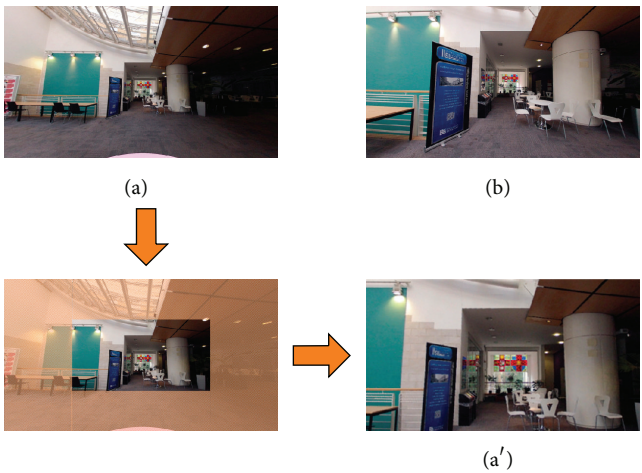


FIGURE 11: (a) Image captured in a route, (b) image captured in front of image (a), and (a') zoom-in of image (a).

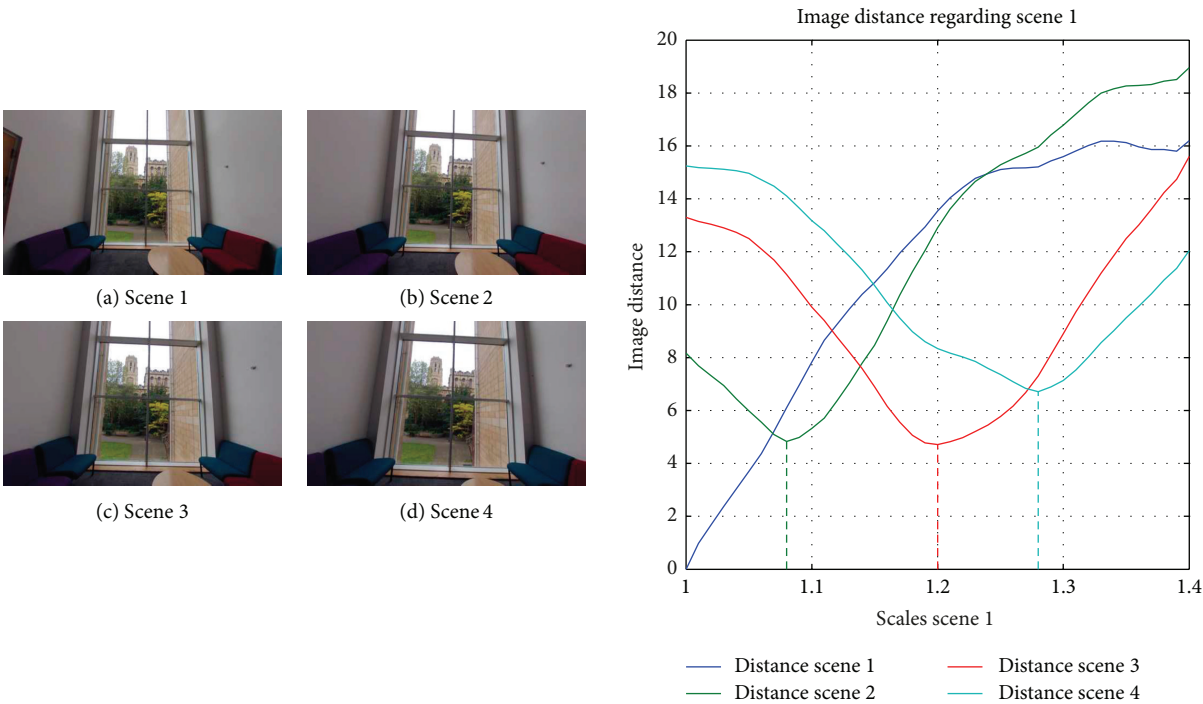


FIGURE 12: Consecutive scenes of a route of images and image distance of the scenes regarding different scales of Scene 1.

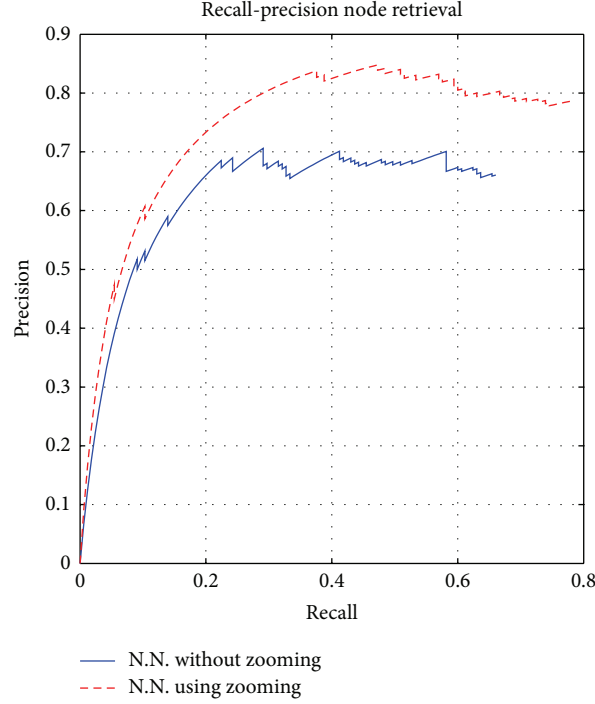


FIGURE 13: Recall-precision of the nearest neighbour in node retrieval using Gist-Gabor as a descriptor. Comparison with and without the multiscale analysis.

the routes' and the nodes' images. This association is used to decide whether a new node is included in the map and is based on the image distance using the global appearance of the scene.

The algorithm can be summarized as follows.

- (i) First, we create the map retrieval database. For that purpose, the algorithm computes the descriptors $z^n \in \mathfrak{R}^{1 \times y}$ of the node imagery (including different scales of every image). y denotes the number of components of each descriptor.
- (ii) The descriptors are stored in columns of a matrix, which represents the map database, $\mathbf{Z} = [z_1^n, z_2^n, \dots, z_i^n, \dots, z_m^n]$, and m is the number of images included in the database, that corresponds with the product of the number of nodes, orientations per node, and number of zoom scales per image.
- (iii) Since the descriptors are stored following the same order as the database images, it is possible to know the node n , orientation in the node θ , and zoom factor scale s^n of each descriptor included in the database, since they are function of the position of the descriptor in the matrix \mathbf{Z} . Denoting i as the number of column in \mathbf{Z} ,

$$[n, \theta, s_n] = f(i). \quad (10)$$

It should be noted that the order in which the nodes are stored in the database does not provide information about its spatial distribution. The position of the nodes is totally unknown to the system when the algorithm starts.

- (iv) When a new route image arrives, the algorithm computes its descriptor z^r , and it calculates its Euclidean distance d with all the descriptors included in \mathbf{Z} :

$$d_i^r = \sqrt{\sum_{a=1}^y (z_{i,a}^n - z_a^r)^2}, \quad i = 1, \dots, m. \quad (11)$$

- (v) The image distance d_i^r is used as a classifier. The algorithm selects the nearest neighbour and is associated with the minimum distance d the corresponding values of n , θ , and s_n .
- (vi) The algorithm repeats this process using different scales of the route image (s_r).
- (vii) Once we have estimated the image association for the different scales s^r , we order the results regarding d and select the experiment with the minimum distance.
- (viii) Finally, we save the parameters corresponding to the minimum image distance (d). From every route image, we obtain the information vector:

$$[n \ d \ \theta \ s_n \ s_r]. \quad (12)$$

5.3. Graph Creation. The process of including a new node in the map starts with the information vector described in (12). We obtain a vector from every route image, and as we study a new image, we add the new information vector to an array. The decision of including a node in the map involves the last 5 route images.

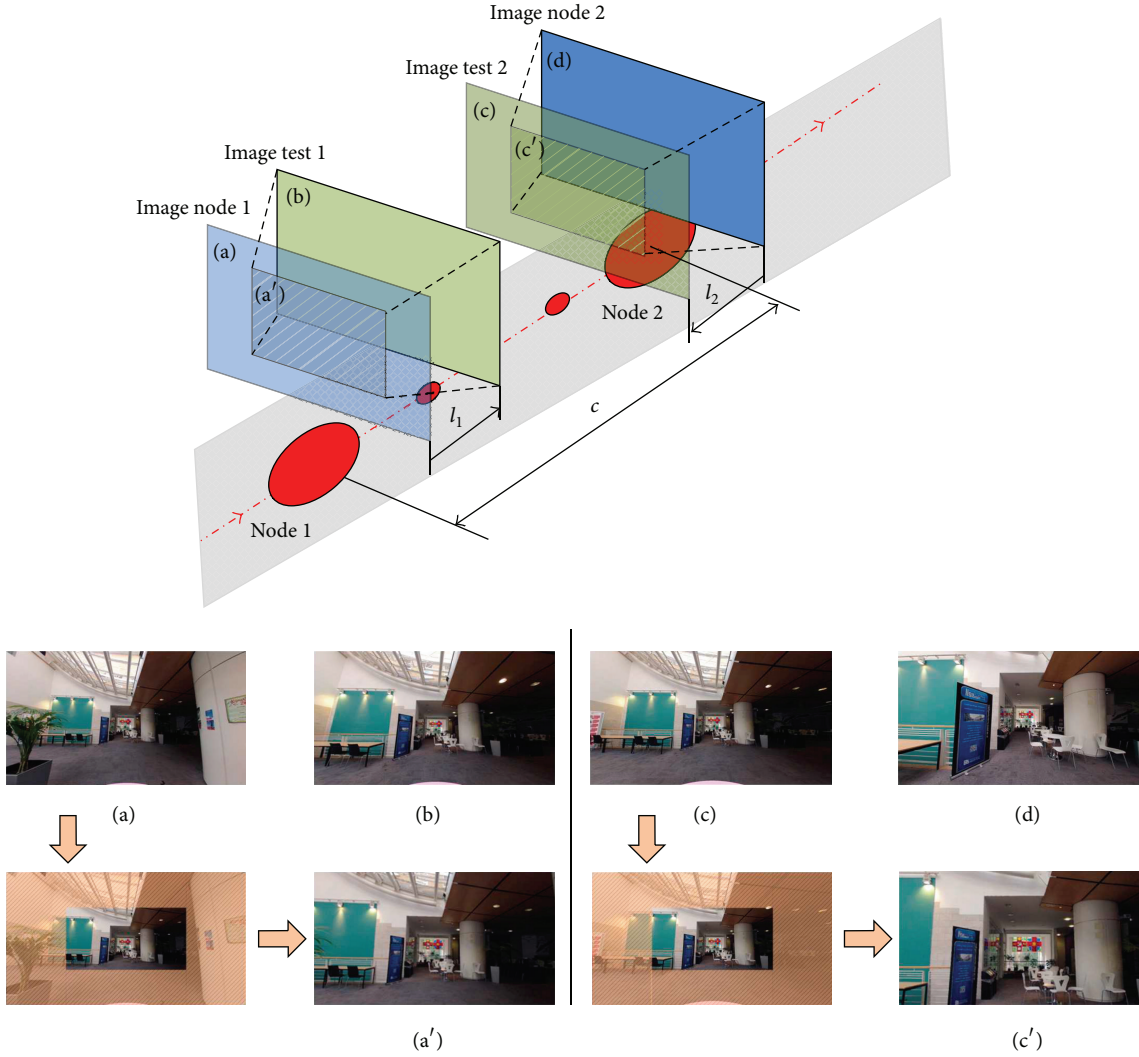


FIGURE 14: (a) Node image, (a') zoom-in of the node image (a), and (b) route image located in front of the node image (a). (c) Route image, (c') zoom-in of the route image (c), and (d) nearest node image of (c) located in front of that image. l_1 and l_2 are topological distances between a route image and the nearest node, and c is the topological distance between nodes.

M is the number of repetitions of the mode value n_m of the nodes number (n) included in the last 5 node's retrieval, and μ and σ are the mean and standard deviation of all the d included in the information vectors so far. The node n_m is included in the graph if any of these two conditions is achieved:

- (i) $M \leq 3$,
- (ii) $M = 2$ and $d_{n_m} < \mu - \sigma$.

Therefore, the algorithm includes a new node if it associates the same node in 3 of the 5 last route images or in 2 of them but with a low image distance (what denotes a highly reliable association).

When the image association has an image distance value $d > \mu + 2\sigma$, the information vector is not taken into account, since a high value of d indicates low reliability in the association.

To know the connections between nodes, we create the adjacency matrix $A \in \mathbb{R}^{N \times N}$, being N the number of nodes. A is a sparse matrix with rows labelled by graph nodes, with 1 denoting adjacent nodes, or 0 on the contrary. Supposing we have included the node n_1 in the graph and the next node found is n_2 , $A_{n_1, n_2} = 1$.

Regarding the topological distance between the nodes in the graph, we make use of the image scale factors. To estimate the distance between two consecutive nodes, the algorithm uses the following information: the topological distance of the last route image in which the first node is detected (l^I), and the scale difference between the first image of the route matched with the next node (l^J). It is worthy of recalling that, as stated above, since the last route scene where a node is detected is due to be in front of that node, the value of l^I will be positive. On the contrary, as the first route image where a new node is matched is usually behind the node, l^J is expected to be negative. So then, the topological distance $c_{n_i, n_{i+1}}$ between






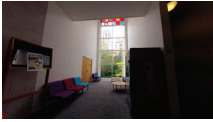
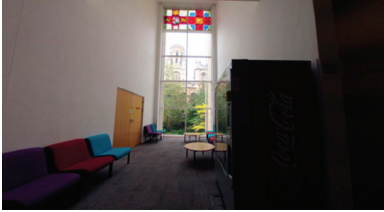
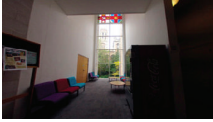
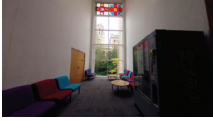


Node images	Route images	Loc. results
$n = 1$ 		$n = 1$ $s_n = 1$ $s_r = 1.35$ $l = -0.35$
		$n = 1$ $s_n = 1$ $s_r = 1.1$ $l = -0.1$
		$n = 1$ $s_n = 1.2$ $s_r = 1.05$ $l = 0.15$
		$n = 1$ $s_n = 1.7$ $s_r = 1.35$ $l = 0.35$
		$n = 1$ $s_n = 1.7$ $s_r = 1.15$ $l = 0.55$
$n = 2$ 		$n = 2$ $s_n = 1.2$ $s_r = 1.35$ $l = -0.15$
		$n = 2$ $s_n = 1$ $s_r = 1$ $l = 0$
		$n = 2$ $s_n = 1.2$ $s_r = 1.1$ $l = 0.1$
		$n = 2$ $s_n = 1.2$ $s_r = 1$ $l = 0.2$

FIGURE 15: Example of image retrieval experiments carried out using two route images and nine scenes of the route that connects both nodes. In the right side, the localization results are shown, including the nearest image (n), the node scene scale factor (s_n), the route scene scale factor (s_r), and the relative topological distance between the node and the route images (l).

a node n_i and n_{i+1} takes into account the sign of the distances regarding the relative position of the route images and the nodes, and it is defined as

$$c_{n_i, n_{i+1}} = l_{n_i}^l - l_{n_{i+1}}^f. \quad (13)$$

Following the example included in Figure 15, l_1^l corresponds with the topological distance obtained in the fifth route image (the last one where the node 1 is detected) and l_2^f with the topological distance of the sixth scene (the first where the node 2 is retrieved). Then, $c_{1,2} = 0.55 - (-0.15) = 0.7$.

To build the graph, it is necessary to incorporate information about the orientation. We suppose that the routes follow a straight path until we detect a change of direction in one of the nodes. $\theta_{n_i}^f$ denotes the orientation associated with the first route image where the node i is retrieved, and the output angle $\theta_{n_i}^l$ is the direction of the last image where the same node is detected. The difference of these angles provides a phase lag that coincides with the change in the direction of the graph:

$$\Delta\theta_{n_i} = \theta_{n_i}^l - \theta_{n_i}^f. \quad (14)$$

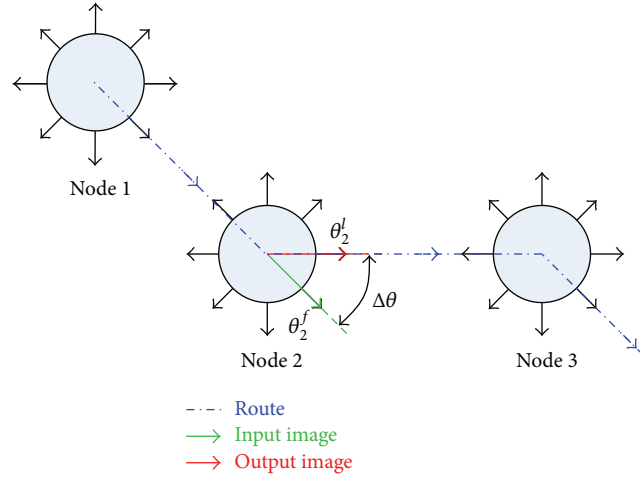


FIGURE 16: Phase change estimation in a node. θ_2^f is the direction of the first image retrieved of node 2 and θ_2^l is the direction of the last image retrieved of node 2. $\Delta\theta$ is the phase lag.

Moreover, $\theta_{n_i}^l$ is the direction the robot has to follow in order to arrive from node n_i to the node n_{i+1} . Figure 16 illustrates the phase lag between two nodes.

We set the orientation of the map by defining the direction of the output image in the first node. That direction determines the global orientation system of the map. The orientation of the graph is updated in every node with the phase lag defined in (14). Since we have a global direction orientation system defined, we can compute for each node the difference of orientation between its local system and the global. For instance, if the input direction of a node is 0° in the global system, and it corresponds to 90° regarding the local system, we have a phase lag of 90° for that node. That way, we can include the new nodes and orientate them according to our global reference system.

When a new route is studied, the algorithm initializes a new coordinates system for its nodes. That route will be analysed independently of the global graph until a common node is found. Using the position and orientation of the common node regarding both systems, we are able to add the new nodes of the current route to the global graph. If two routes share a common path and we match nodes of the map database that have been previously included in the global map, the topological distances $c_{n_i, n_{i+1}}$ between those common nodes are estimated again, and the results are taken into account in the graph by calculating the mean of the new estimation, $c_{n_i, n_{i+1}}$, with the previous estimations. The mean will be weighted by the number of times that the same distance has appeared.

Therefore, our map building algorithm takes advantage of the information provided by the routes in order to obtain the relative position of the nodes by matching the sequence of images with the nodes descriptors database. It uses that information to estimate the adjacency relations, but it also gives information about the relative distance and position between them using the multiscale analysis.

6. Path Estimation Algorithm

Once we have carried out the map building and we obtain the graph that represents the layout of the nodes, our aim is to estimate the path of the routes that the robot follows during the navigation in this graph. We can divide the localization of the robot in the map in two main steps: first, we carry out a coarse estimation, identifying the nearest node and the orientation. Note that the orientation of the robot is determined using the phase of the node image associated with the route image. The algorithm uses a weighting function to penalize associations that are geometrically far from the previous route pose, since consecutive route images should be located nearby in the graph.

If the localization of the route images is based only on the matching with the nodes in an image retrieval process, the localization accuracy will be limited to the node positions. In order to obtain a more accurate estimation of the pose, the second step in the localization algorithm includes the multiscale analysis. Specifically, we apply this technique using the current route image and the associated node image. That way, the algorithm is able to find the relative position between both images and to extend the possible position values to intermediate position of the nodes locations.

When a route image arrives, we compare several zoom scales of this image with the nodes database, \mathbf{Z} , that includes the descriptor of different scales of the nodes images. The association between the route image and the database is determined again using the nearest neighbour regarding the image distance (d). Since the test images come from a route path, we can suppose that the distance and phase lag between consecutive images should be not high. For that reason, in order to improve the localization of the route images, the algorithm introduces a weighting function in order to penalize the probability of finding the current location or orientation far away from the previous image pose.

6.1. Weighting Function. As stated at the beginning of this section, we introduce a weighting function in the algorithm to improve the localization accuracy of the route images in the topological map. This function reduces the probability of finding the location of the current nearest node distant from the previous image pose. Since the image association criteria are the nearest neighbour, the weighting function increases the image distance of the associations whose node image pose is distant from the last robot pose. In this way, we reduce the likelihood of selecting them as the current nearest node.

The weighting function is composed of 2 terms: the first one takes into account the topological distance between consecutive route images in the graph and the second their phase lag.

The first term uses the positions of the nearest nodes associated with the previous and the current route images in order to estimate their topological distance in the map. The adjacency matrix A allows us to find out the shortest path between two nodes in the map. Since we have a connected graph, we can always find a path that connects any 2 nodes of the map. c_{n_1, n_2} represents the cost of traversing 2 adjacent nodes $n_1, n_2 \in A$ (that corresponds with the topological distance between nodes, defined in (13)) and $P_{n_i, n_j} = [n_i, \dots, n_j]$ the sequence of nodes of the shortest path that connects n_i and n_j , the cost C_{n_i, n_j} associated with the sequence of nodes P_{n_i, n_j} can be defined as

$$C_{n_i, n_j} = \sum_{n_i}^{n_j} c_{n_i, n_{i+1}}. \quad (15)$$

The second term takes into account the phase lag between consecutive route poses.

Finally, the weighting function between two images can be defined as

$$w(n_i, n_j, \theta_{n_i}, \theta_{n_j}) = w_1 \cdot C_{n_i, n_j} + w_2 \cdot |\theta_{n_j} - \theta_{n_i}|, \quad (16)$$

where w_1 and w_2 are constants that module the weight action of the topological distance and the phase lag, respectively.

As (16) shows, the weighting value between 2 images depends on the cost to traverse the path that connects their respective closest nodes and their orientation difference.

6.2. Route Images Localization in the Graph. First, the algorithm computes the image distance between the current route image and the nodes images using (11). From it, we obtain d_i^r , $i = 1, \dots, m$, that represents the image distance of the route image with every image included in the map database, \mathbf{Z} . It includes the descriptors of all the nodes images with different scales.

Since the map database includes different zoom scales of image of the nodes, each descriptor included in \mathbf{Z} has a value of n , θ , and s_n associated (10). The algorithm compares the current route image descriptor with \mathbf{Z} , classifies the results regarding the value of image distance d , and selects the k -nearest neighbours. Then, this process is repeated using different zoom scales of the route image s_r .

After that, we update the image distance values d of the k neighbours selected using each scale s_r using the weighting function:

$$d' = d \times w(n_i, n_{i-1}, \theta_{n_i}, \theta_{n_{i-1}}), \quad (17)$$

with n_{i-1} and $\theta_{n_{i-1}}$ the nearest node and orientation of the previous route image and n_i and θ_i the nearest node and orientation of each neighbour selected in the matchings. The weighting value may change for every neighbour, since n_i and θ_i might be different in every particular case.

When all the image distances have been updated, we classify again the results regarding d' and choose the Nearest Neighbour. With the information associated with the retrieval, we find out the closest node n of the route image, its orientation θ , and the scale factors of both the node and the route images (s^n and s^r).

With this data, we can determine the current robot pose in the map. The position is estimated using the nearest node, and the relative position between the matched images, provided by the multiscale analysis and the difference of the scale factors s_n and s_r defined in (9). The direction of advance is provided by θ . Note that θ is the orientation of the node image regarding the local reference system of each node that must be corrected with the phase lag between the map global system and the node reference system, estimated previously during the map building process.

7. Experiments and Results

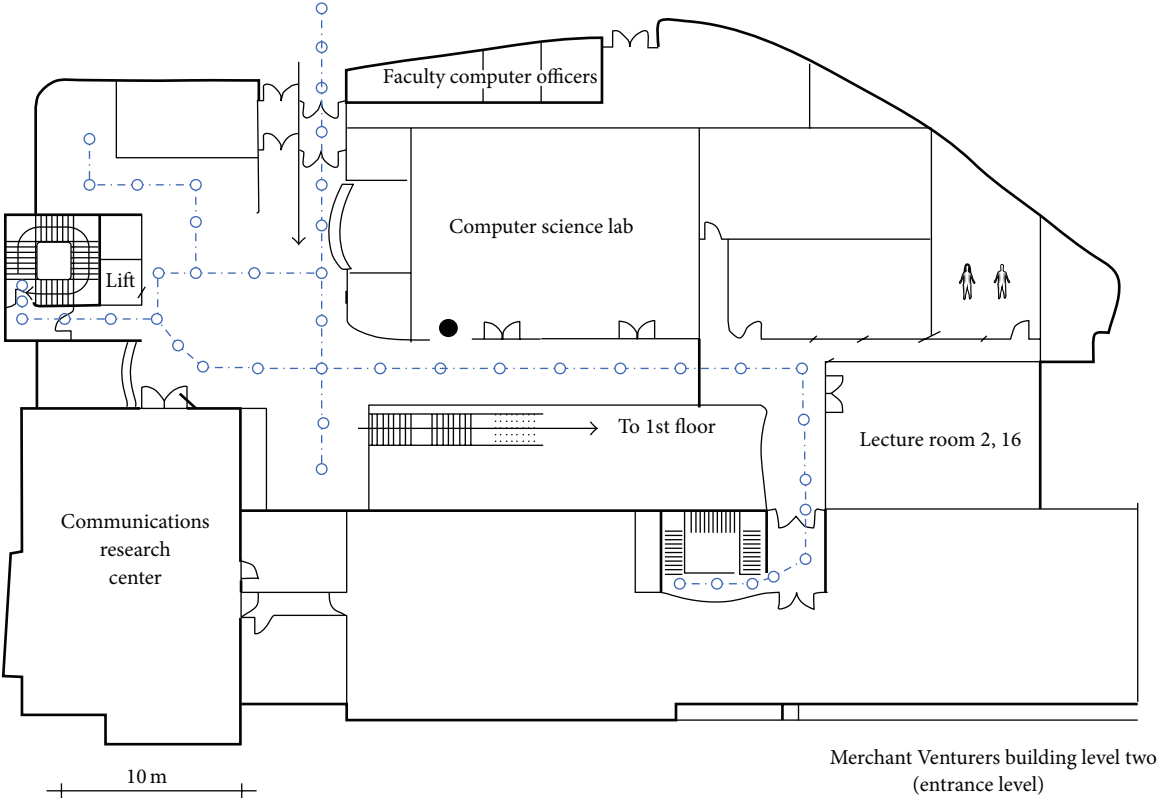
This section details the database used during the experiments and the results of the map building and route path estimation using the multiscale analysis and the global appearance of images. As stated at the end of Section 3, the technique selected to describe the global appearance of the images is Gist-Gabor, and the image resolution is 32×64 pixels.

7.1. Dataset: Nodes and Routes Images. Two different databases have been captured. They correspond to common areas of the Merchant Venturers Building of the University of Bristol.

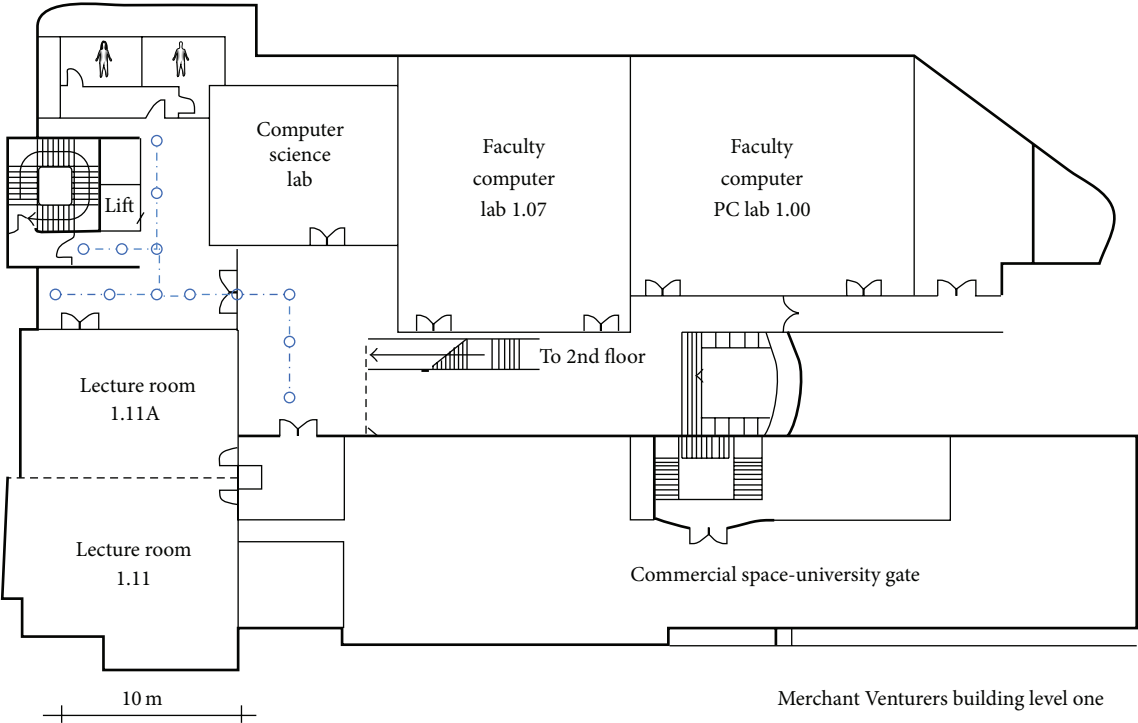
Each database is composed of a set of nodes, and different routes of images are distributed along the areas where the nodes were captured. Note that each node has 8 images, with a phase lag of 45 between consecutive images, covering the complete file of view around the position of the node.

The experiments are divided into two different areas. The number of images of each area appears in Table 2, and the real distribution of the nodes in Figure 17. The actual distance between consecutive nodes is 2 meters as a rule, but in places where an important change of appearance is produced, that is, changes of direction or crossing a door, a new node is captured independently of the distance with the previous node. For that reason, the distance may be lower.

Regarding the routes, the frequency of image acquisition is higher in the routes of Area 2. The images are taken every 0.5 meter in Area 1 and every 0.2 meter in Area 2. We increase the capture rate at turnings. We take a minimum of four images per position when a change of orientation is produced.



(a) Area 1



(b) Area 2

FIGURE 17: Representation of the graphs in the plane of each navigation area.

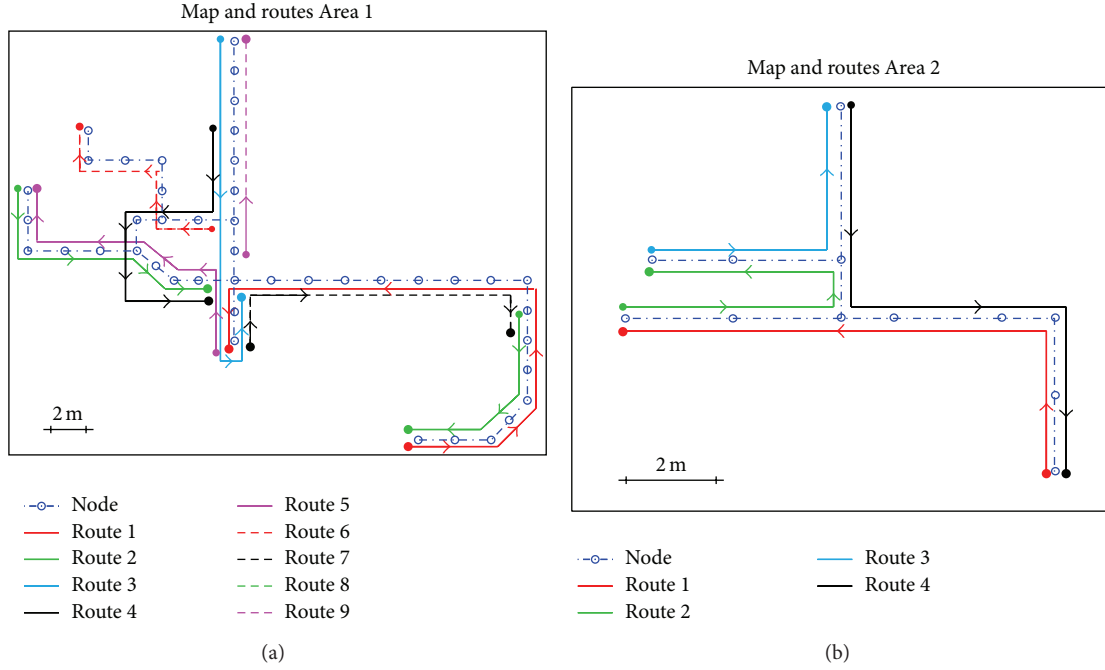


FIGURE 18: Synthetic distribution of the nodes and routes for (a) Area 1 and (b) Area 2.

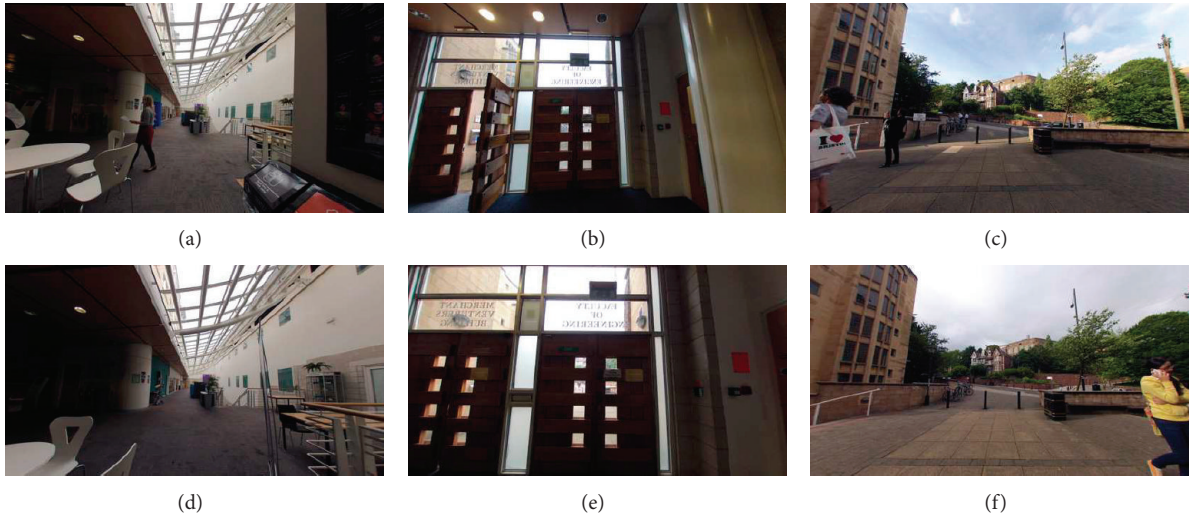


FIGURE 19: Examples of images used in the experiments: (a), (b), and (c) are routes images and (d), (e), and (f) are the corresponding nearest nodes images.

In Area 2, this frequency increases with a minimum of 6 images per position. Figure 18 shows the distribution of the nodes and the routes in a synthetic representation. Figure 19 presents some examples of node and route images. They show typical situations of real applications, such as changes in illumination conditions and movements of the furniture and occlusions produced by people moving in the area. The system must be able to cope with these situations.

7.2. Map Building Results. Figure 20 presents the nodes graph of (a) Area 1 and (b) Area 2. It has been obtained after running our algorithm. We can appreciate that the algorithm is able to estimate the connections between nodes, with a similar

distribution regarding the real layout in the both areas. Area 1 has been the most challenging due to the higher number of nodes, the transition from different rooms, and the loop closure in the map. In the loop closure, the graph representation slightly differs from the real layout. However, although the map loses some accuracy, the navigation of the robot is not affected. The robot can navigate from one node to the other by knowing the node output image that connects the first node to the second one, and this does not depend on the graph layout.

It is important to remark that the algorithm needs a minimum number of route images between nodes. Otherwise, a node might not be included in the map.

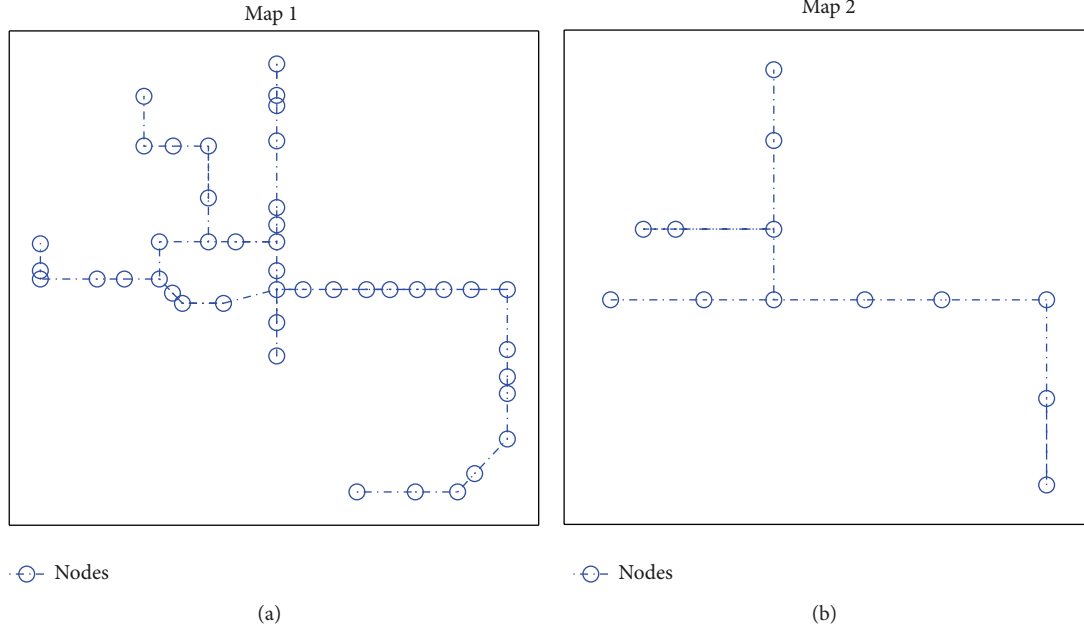


FIGURE 20: Graph representation of the nodes arrangement obtained with the map building algorithm for (a) Area 1 and (b) Area 2.

TABLE 2: Number of images per area.

	Number of images (Area 1)	Number of images (Area 2)
Nodes	352	52
Route 1	110	100
Route 2	50	72
Route 3	67	66
Route 4	58	125
Route 5	62	—
Route 6	46	—
Route 7	69	—
Route 8	67	—
Route 9	40	—

TABLE 3: Procrustes analysis results of the graphs obtained in the map building.

	Area 1	Area 2
μ	0.0372	0.0078

We use the Procrustes analysis [41, 42] in order to measure the error of the graphs obtained in the map building. This analysis studies the geometric error between the real layout of the nodes and the layout obtained with our algorithm. It returns a standardized value of dissimilarity $\mu \in [0, 1]$. The lower μ , the more accurate graph. We show the results in Table 3. In both cases, the geometric error is considerably low.

The system is especially sensitive in the phase lag between nodes, since it is based on the angle estimation of the input and output images of the node. For that reason, it is advisable to raise the frequency of the image acquisition in the nodes where there is a change of direction. In the experiments,

the maximum value of s^n is 2.5, with a fixed step of 0.1 between consecutive scale factors. Regarding the routes images, s^r has a maximum value of 1.4, with a step of 0.05. We have chosen a small step in both route and node scales since we have given priority to the performance of the results over the computational requirements. The average time per image in Area 1 is 725 ms and in Area 2 is 680 ms. The difference of time requirements is due to the matching of the routes images with the nodes database, since the number of nodes in Area 1 is bigger. The estimation of the Euclidean distance between the new route image descriptor and the descriptors contained in the database supposes the 45% of the total time in the map building process. Area 1 contains more nodes than Area 2, so that its descriptors database has a higher number of elements, what supposes more time to carrying out the retrieval and, therefore, an increase of the global process time in the map building.

The orientation of the global reference system is determined by defining the direction of the output image of the first node. In the experiments, we choose this direction so that the graph has the same orientation that the layout represented in Figure 18. If we had chosen any other direction, the map shape would have been the same, but rotated. Anyway, the orientation of the global reference system does not affect the localization algorithm.

7.3. Path Estimation Results. In order to find proper values of the weighting constants, we carry out a study of the localization performance regarding the values of w_1 and w_2 . Figure 21 shows the precision in the node image retrieval varying both parameters. The dataset of the experiments is composed of the images included in the routes 1 and 5 of the map 1. In the precision measurement, we consider that a retrieval has been successful when it selects the node image that corresponds

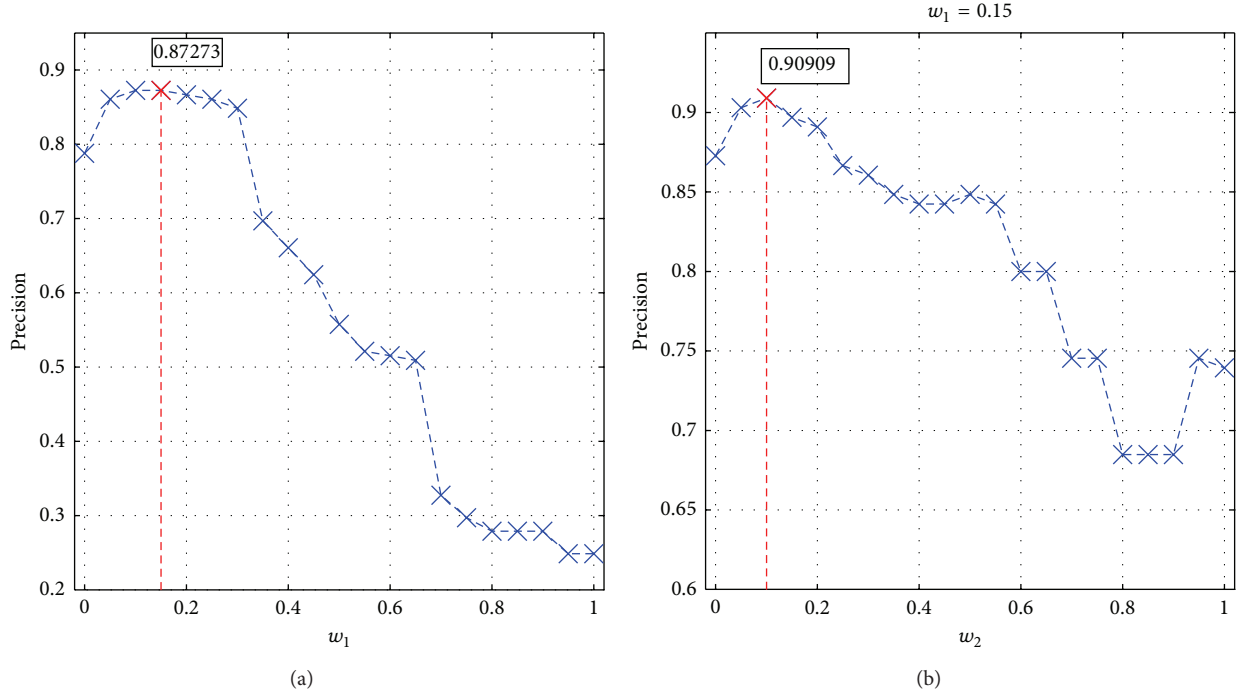


FIGURE 21: Image retrieval precision regarding the image distance weighting parameters. (a) Precision varying the topological distance constant (w_1) and (b) precision varying the phase change constant (w_2).

with both the correct position and orientation. In Figure 21(a) we study the retrieval performance regarding the weighting constant w_1 . We can notice an increase of the precision for low values of w_1 . Once we have selected w_1 , we study the precision varying the parameter w_2 . The results are shown in in Figure 21(b). Both graphs prove that the weighting function improves the retrieval precision. However, if we are too restrictive with the position or phase changes, the precision decreases. For that reason, when the constants are given high values, the retrieval accuracy is lower.

In the path localization experiments, the weighting constants are given the values $w_1 = 0.15$ and $w_2 = 0.1$, and we use $k = 10$ nearest neighbours when doing the retrieval of each zoom scale of the route images. The node zoom scale s_n varies from 1 to 2.2 with a step of 0.4. Regarding the route zoom scale, it varies from 1 to 2.2 with a step of 0.3 between consecutive scales.

Figure 22 shows the path estimation of different routes of both areas. The dots in the paths of the routes represent the position of the different images studied. As it can be seen, the algorithm copes with the interpolation of the location in halfway positions between the nodes using the image's scales information. In general, the precision at turnings in the routes decreases. It is also important that, despite the fact that we introduce the weighting function, the algorithm is able to find again the correct position although a previous estimation is not correct, as we can see in Figures 22(a) or 22(c). The result in the path planning of the fourth route of the first area (Figure 22(b)) is also interesting. As we can appreciate in Figure 18(a), the route number 4 presents a variation in its path that differs from the layout of the nodes. However,

despite that fact, the path estimation algorithm is able to estimate the position accurately.

Therefore, the results prove that our algorithm is able to estimate the path of the route even in intermediate positions of the nodes and deal with the correction of false association of nodes in previous parts of the route.

8. Conclusions

In this paper we have studied the problem of the only-visual topological mapping and route navigation using global appearance image descriptors. First, we have included a comparison of three global appearance techniques and different image sizes. Next, we present the multiscale analysis, which permits estimating the relative position of two images using digital zooming. Then, we include an algorithm to build a topological map from a set of nodes and routes of images. Finally, we have developed a localization algorithm that estimates the position of the mobile in the graph using the visual information as input.

In the comparison of the global appearance descriptors, all the techniques show a reasonable high accuracy in image retrieving tasks. Fourier analysis presents a stable retrieving precision with regard to the images size and a reduced time requirement, but the memory requirement is clearly higher than the other techniques, especially with the bigger images. HOG descriptor shows good computational cost and memory requirements. However, when we reduce the image size, the accuracy in localization decreases more than in the case of the other techniques. Hence, since we pretend to use a reduced image resolution in the map building

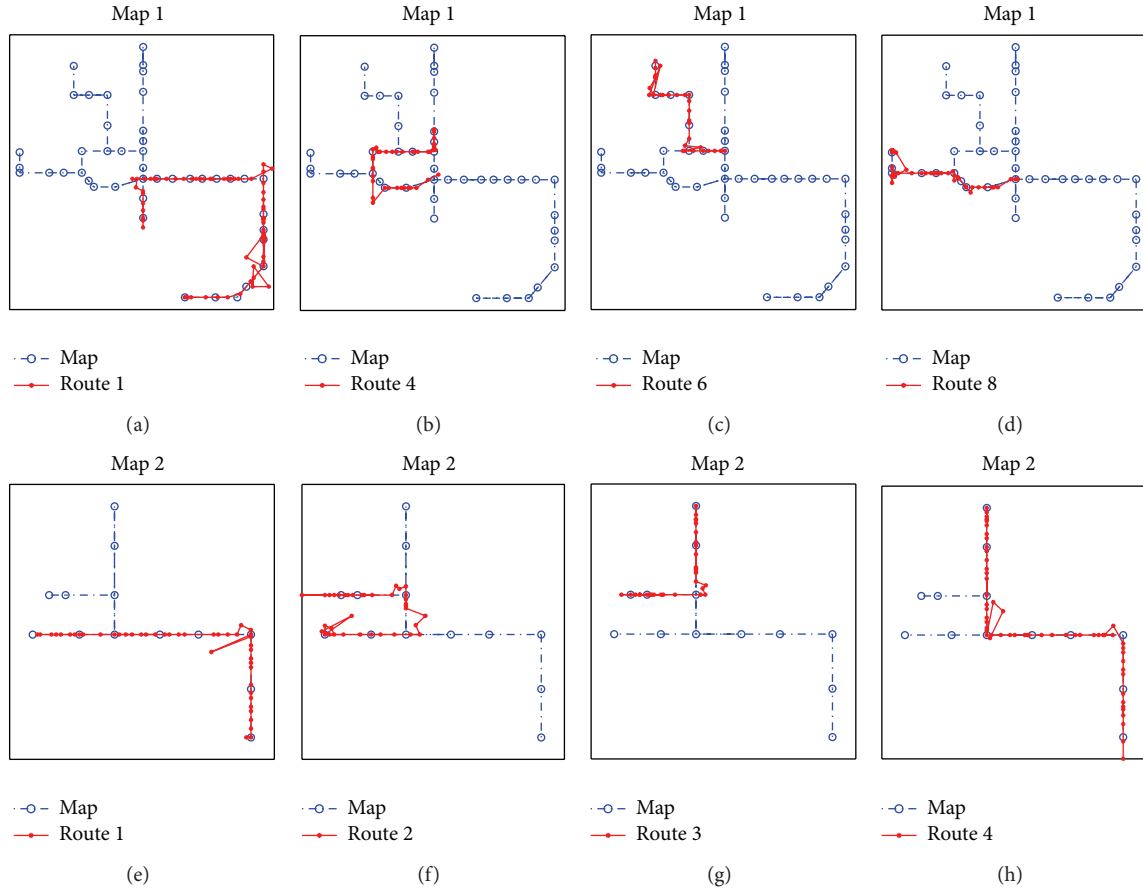


FIGURE 22: Path estimation of the (a) Route 1, (b) Route 4, (c) Route 6, and (d) Route 8 of the Area 1 and path estimation of the (e) Route 1, (f) Route 2, (g) Route 3, and (h) Route 4 of Area 2.

and localization algorithms, HOG is inadvisable in these applications. Gist-Gabor is the most compact representation in almost all the experiments. We select this descriptor to carry out the experiments due to the fact that it is the most reliable descriptor using the lower image resolutions. It is possible to reduce the scene almost 30 times the original size without an important detriment of precision. In this way, the computational time in the image processing in order to obtain the descriptor is reduced more than 10 times.

Regarding the multiscale analysis, it improves the association between images using the global appearance of the scenes and provides a measurement of the topological distance between images.

The map building algorithm is able to determine the adjacency relationships between the nodes distributed in the navigation area and to create a graph using the information of routes taken along the nodes positions. The results present a high accuracy in the node detection and estimation of adjacency and relative orientation. Moreover, the estimation of the topological distance between the nodes provides a graph representation of the nodes with similar layout to the real distribution.

The algorithm created to estimate the path of routes along the area takes advantage of the multiscale analysis to improve the topological localization of the robot in the map.

After doing the matching of the route image with the map database, the difference of scales between the node and the route image provides the relative position of both images. Although we use a weighting function in order to penalize important changes in position and orientation between consecutive route images, the algorithm is able to find again the correct location although a previous image of the route would introduce a false pose.

The results obtained both in the map building and the path representations of routes encourage us to continue the possibilities of the application of global appearance image descriptors to these tasks. It would be interesting to extend this study to find the minimum information that the map has to include in order to allow a correct navigation of the robot, the application of new global appearance descriptors, the use of omnidirectional visual information, or the improvement in the estimation of the orientation in order to correct small errors during the navigation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been supported by the Spanish government through the Project DPI2010-15308 “Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad.”

References

- [1] R. Sim and G. Dudek, “Effective exploration strategies for the construction of visual maps,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 4, pp. 3224–3231, October 2003.
- [2] T. Camus, D. Coombs, M. Herman, and T.-H. Hong, “Real-time single-workstation obstacle avoidance using only wide-field flow divergence,” in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 323–330, Vienna, Austria, August 1996.
- [3] S. Badal, S. Ravela, B. Draper, and A. Hanson, “A practical obstacle detection and avoidance system,” in *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, pp. 97–104, Sarasota, Fla, USA, December 1994.
- [4] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, “Omni-directional vision for robot navigation,” in *Proceedings of the IEEE Workshop On Omnidirectional Vision*, pp. 21–28, 2000.
- [5] H. Morita, M. Hild, J. Miura, and Y. Shirai, “Panoramic view-based navigation in outdoor environments based on support vector learning,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 2302–2307, Beijing, China, October 2006.
- [6] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [7] A. Lingua, D. Marenchino, and F. Nex, “Performance analysis of the sift operator for automatic feature extraction and matching in photogrammetric applications,” *Sensors*, vol. 9, no. 5, pp. 3745–3766, 2009.
- [8] A. C. Murillo, J. J. Guerrero, and C. Sagüés, “SURF features for efficient robot localization with omnidirectional images,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3901–3907, April 2007.
- [9] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [10] R. Gartshore, A. Aguado, and C. Galambos, “Incremental map building using an occupancy grid for an autonomous monocular robot,” in *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARV '02)*, vol. 2, pp. 613–618, December 2002.
- [11] B. Kröse, R. Bunschoten, S. ten Hagen, B. Terwijn, and N. Vlassis, “Household robots look and learn: environment modeling and localization from an omnidirectional vision system,” *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 45–52, 2004.
- [12] E. Menegatti, T. Maeda, and H. Ishiguro, “Image-based memory for robot navigation using properties of omnidirectional images,” *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.
- [13] I. Kunttu, L. Lepistö, J. Rauhamaa, and A. Visa, “Multiscale fourier descriptor for shape-based image retrieval,” in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 765–768, August 2004.
- [14] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116–121, March 1985.
- [15] A. Gil, Ó. Reinoso, M. Ballesta, M. Juliá, and L. Payá, “Estimation of visual maps with a robot network equipped with vision sensors,” *Sensors*, vol. 10, no. 5, pp. 5209–5232, 2010.
- [16] J. Gaspar, N. Winters, and J. Santos-Victor, “Vision-based navigation and environmental representations with an omnidirectional camera,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 890–898, 2000.
- [17] N. Winters and J. Santos-Victor, “Omni-directional visual navigation,” in *Proceedings of the 7th International Symposium on Intelligent Robotics Systems*, pp. 109–118, 1999.
- [18] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor, “Visual servoing and appearance for navigation,” *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 87–97, 2000.
- [19] J. Košecká, L. Zhou, P. Barber, and Z. Duric, “Qualitative image based localization in indoors environments,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II-3–II-8, June 2003.
- [20] A. Štimec, M. Jogan, and A. Leonardi, “Unsupervised learning of a hierarchy of topological maps using omnidirectional images,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 4, pp. 639–665, 2008.
- [21] J. Choi, M. Choi, and W. K. Chung, “Topological localization with kidnap recovery using sonar grid map matching in a home environment,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 366–374, 2012.
- [22] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, “The role of homing in visual topological navigation,” in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 567–572, Vilamoura, Portugal, October 2012.
- [23] M. Cummins and P. Newman, “FAB-MAP: probabilistic localization and mapping in the space of appearance,” *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [24] M. Cummins and P. Newman, “Appearance-only SLAM at large scale with FAB-MAP 2.0,” *International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.
- [25] N. Bellotto, K. Burn, E. Fletcher, and S. Wermter, “Appearance-based localization for mobile robots using digital zoom and visual compass,” *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 143–156, 2008.
- [26] M. Milford, G. Wyeth, and D. Prasser, “Simultaneous localization and mapping from natural landmarks using ratslam,” in *Proceedings of the Australasian Conference on Robotics and Automation*, N. Barnes and D. Austin, Eds., Australian Robotics and Automation Association, Canberra, Australia, 2004.
- [27] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, “Fab-map + ratslam: appearance-based slam for multiple times of day,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 3507–3512, May 2010.
- [28] M. Milford, “Visual route recognition with a handful of bits,” in *Proceedings of the Robotics Science and Systems Conference*, N. Roy, Ed., University of Sydney, Sydney, Australia, 2012.
- [29] Woodman Labs Inc, 2013, <http://gopro.com/hd-hero2-cameras/>.

- [30] L. Payá, L. Fernández, Ó. Reinoso, A. Gil, and D. Úbeda, "Appearance-based dense maps creation: comparison of compression techniques with panoramic images," in *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics*, pp. 250–255, Milan, Italy, July 2009.
- [31] F. Amorós, L. Payá, O. Reinoso, and L. M. Jiménez, "Comparison of global appearance techniques applied to visual map building and localization," in *International Conference on Computer Vision Theory and Applications (VISAPP '12)*, vol. 2, pp. 395–398, SciTePress, Science and Technology Publications, Rome, Italy, February 2012.
- [32] L. Payá, F. Amorós, L. Fernández, and O. Reinoso, "Performance of global-appearance descriptors in map building and localization using omnidirectional vision," *Sensors*, vol. 14, no. 2, pp. 3033–3064, 2014.
- [33] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of the 4th IEEE International Conference on Computer Vision Systems (ICVS '06)*, p. 45, January 2006.
- [34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, June 2005.
- [35] A. Friedman, "Framing pictures: the role of knowledge in automatized encoding and memory for gist," *Journal of Experimental Psychology: General*, vol. 108, no. 3, pp. 316–355, 1979.
- [36] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [37] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.
- [38] D. Gabor, "Theory of communication. Part III: radio and communication engineering," *Journal of the Institution of Electrical Engineers*, vol. 93, no. 26, pp. 429–457, 1946.
- [39] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.
- [40] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM," *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.
- [41] D. G. Kendall, "A survey of the statistical theory of shape," *Statistical Science*, vol. 4, no. 2, pp. 87–99, 1989.
- [42] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*, Wiley Series in Probability and Statistics: Probability and Statistics, John Wiley & Sons, Chichester, UK, 1998.

