applied sciences

MDPI

Article

Creating Incremental Models of Indoor Environments through Omnidirectional Imaging

Vicente Román *,†,‡, Luis Payá ‡, Sergio Cebollada ‡ and Óscar Reinoso ‡

Department of Systems Engineering and Automation, Miguel Hernández University, 03202 Elche, Spain; lpaya@umh.es (L.P.); sergio.cebollada@Umh.es (S.C.); o.reinoso@umh.es (Ó.R.)

* Correspondence: v.roman@umh.es; Tel.: +34-96-665-2435

† Current address: Avda. de la Universidad, s/n. Ed, Innova, 03202 Elche, Spain.

‡ These authors contributed equally to this work.

Received: 3 August 2020; Accepted: 13 September 2020; Published: 17 September 2020

Abstract: In this work, an incremental clustering approach to obtain compact hierarchical models of an environment is developed and evaluated. This process is performed using an omnidirectional vision sensor as the only source of information. The method is structured in two loop closure levels. First, the Node Level Loop Closure process selects the candidate nodes with which the new image can close the loop. Second, the Image Level Loop Closure process detects the most similar image and the node with which the current image closed the loop. The algorithm is based on an incremental clustering framework and leads to a topological model where the images of each zone tend to be clustered in different nodes. In addition, the method evaluates when two nodes are similar and they can be merged in a unique node or when a group of connected images are different enough to the others and they should constitute a new node. To perform the process, omnidirectional images are described with global appearance techniques in order to obtain robust descriptors. The use of such technique in mapping and localization algorithms is less extended than local features description, so this work also evaluates the efficiency in clustering and mapping techniques. The proposed framework is tested with three different public datasets, captured by an omnidirectional vision system mounted on a robot while it traversed three different buildings. This framework is able to build the model incrementally, while the robot explores an unknown environment. Some relevant parameters of the algorithm adapt their value as the robot captures new visual information to fully exploit the features' space, and the model is updated and/or modified as a consequence. The experimental section shows the robustness and efficiency of the method, comparing it with a batch spectral clustering algorithm.

Keywords: mapping; incremental clustering; mobile robots; global-appearance descriptors; omnidirectional images; computer vision

1. Introduction

Presently, the use of visual information in mobile robotics is widely expanded. Independent of the final task it was designed for, an autonomous mobile robot must solve continuously two crucial problems: it has to build a model of the environment (mapping) and to estimate the position of the robot within this model (localization). Both critical problems should be solved with an acceptable accuracy and computational cost.

Mapping was facilitated by the continuous progress of mobile robots' abilities in perception and computation, which enable them to improve their operability in large and heterogeneous zones without the necessity of introducing changes into the environment structure. Two main frameworks were used in order to carry out the mapping task: the metric maps [1]; and the topological maps [2].

Regarding the topological maps, some related works propose arranging the information hierarchically, into several layers with different levels of granularity [3,4].

Hierarchical maps constitute a convenient framework to arrange the information. Notwithstanding, visual mapping remains an important research field in robotics due to the problems that the algorithms may have in heterogeneous and challenging environments. An important alternative to build hierarchical maps is by using some clustering techniques, such as spectral clustering [5]. Some researchers used spectral clustering methods along with visual information to build topological maps [6–9]. Spectral clustering proved to be useful because it can cluster robustly highly dimensional information compared to other well-known methods [10,11].

Even though spectral clustering was used to build maps in mobile robotics with good results [11], the problems of using the standard spectral method in incremental mapping are threefold. Firstly, most spectral clustering methods require the number of final nodes to be indicated previously. Secondly, computing similarities among entities can be hard in terms of computational cost. This is especially noticeable when the data set becomes large. Finally, as a consequence of the previous problem, it is not possible to perform spectral clustering *on-line* when the dataset is constantly growing (i.e., The robot is continuously moving and capturing new images which must be included in the model).

Presently, incremental mapping is a key ability because it would enable mobile robots to gradually build or update a model as they explore the initially unknown environment and capture new information. For this purpose, incremental clustering methods can be useful [12].

In this work, we propose a framework based on incremental clustering with the objective of creating a topological hierarchical map incrementally, using only visual information. Each group of similar images will be included in a cluster with a representative descriptor associated to it. To avoid the necessity of setting the number of clusters beforehand, our proposal defines a set of adaptive thresholds. Depending on them, the algorithm behaves more or less restrictively and therefore creates a higher or lower number of clusters as it progresses. In addition, the method proposed in this paper is able to perform *on-line*, updating the model every time the robot captures a new image.

When using computer vision to build a model of the environment, it is important to verify the performance of the method when the visual appearance of the environment changes substantially. In real-operating conditions the robot has to cope with different events: different lighting conditions during the operation, scenes partially occluded by people or other mobile robots and changes in the scene due, for example to the furniture position. For that reason experiments were carried out in a real environment during working hours.

The remainder of the paper is structured as follows. In the following section, related works on visual information and incremental and hierarchical mapping are outlined. In Section 3, global appearance descriptors are defined. The mapping method is presented in Section 4. In Section 5, relevant information about the datasets, tools and acquisition equipment used in the experiments is presented. In Section 6, the results of the experiments are shown. Finally, conclusions and future research lines are presented in Section 7. An extra section named 'Supplementary Materials' includes some tables of symbols, which summarize all the variables and parameters used to describe the proposed method.

2. Related Works

2.1. Image Description

As explained in the introduction, mobile robots have to solve the mapping and localization problems during their autonomous navigation. To deal with these tasks the robot has to capture relevant information about its movement and the environment, and a variety of tools can be found in the literature to obtain that information and to process it. Encoders permit calculating the odometry of the robot and they are one of the main sensors used in mobile robotics, but the information obtained from the encoders should not be used as an absolute measure because of the error they accumulate. Taking that into

account, in real implementations, odometry data are complemented by other sensors. For instance, it can be complemented with sonars [13] or lasers [14]. In addition, mobile robots can work also with GPS, which presents a good performance in outdoor applications but it has little ability to offer signal in indoor or narrow outdoor zones. Complementing these sensors, visual sensors constitute currently the principal area to investigate new approaches to robot navigation. Cameras offer a lot of information from the environment with a relatively low cost, they can be used in outdoor and indoor environments and they permit solving other high-level tasks such as human identification, face recognition [15] and obstacles detection [16]. There are some examples of sensors combinations: For example Choi et al. [17] present a system that combines sonar with visual information to perform SLAM (Simultaneous Localization And Mapping) tasks. More information about these systems can be found in [18], where a review of these techniques is done. The visual information can be captured with a variety of devices; from a single camera [19], stereo cameras [20] or omnidirectional cameras [21,22].

Solving navigation tasks using only visual information is a challenging problem. Images contain much information and relevant data should be extracted from them in order to ease the mapping and localization tasks. The use of local descriptors was the classical approach for obtaining relevant information from the images and the method consists of extracting outstanding landmarks or regions and describing them using algorithms such as SIFT (Scale-Invariant Feature Transform) [23], SURF (Speeded-Up Robust Features) [24] or BRIEF (Binary Robust Independent Elementary Features) [25]. This is a mature alternative and many researchers make use of it in mapping and localization. For example, Angeli et al. [26] proposed using these descriptors to perform topological localization. Valgren and Lilienthal [7] did that work in outdoor areas. Murillo et al. [27] present a comparative study for the localization task using local-appearance descriptors in indoor environments. A comparative evaluation of this kind of local appearance methods was made by Gil et al. [28]. They evaluated the repeatability, the invariance against small changes and distinctiveness of the descriptors under different perceptual conditions. They proved that the behaviour of local appearance descriptors can deteriorate when unexpected conditions appear. Taking it into account that the information not only changes while the robot is moving and the perspective varies, but also due to changing lighting conditions, occlusions by furniture or people and noise during the acquisition, it is necessary to calculate more robust and invariant descriptors. In this sense, global-appearance descriptors constitute a powerful alternative, which consists of describing each image as a whole, without detecting any landmark or local feature. This method creates a more intuitive representation of the environment, simplifies the navigation process and permits building topological maps of the environment. Since each image is described with a unique vector, localization can be carried out with simple algorithms, based on the pairwise comparison of vectors. Additionally, calculating global-appearance descriptors from omnidirectional images constitutes a powerful approach due to the wide field of view of such images, leading to robust and rotationally invariant descriptors, such as those presented in the next paragraph.

Diverse global appearance methods were studied in recent years. One of these methods is based on the Histogram of Oriented Gradients (HOG). Using this descriptor, diverse problems were solved. For instance Dalal and Triggs [29] explain how to use it to detect pedestrians in a real situation, Paya et al. [30,31] use HOG to solve the localization problems and to create hierarchical maps. Gist is another description method, proposed by Oliva and Torralba in [32] and it was used in works such as [33] where it is described as a biologically inspired vision localization system and the descriptor is tested in different outdoor environments. It was also used by Zhou et al. [34] to solve the localization through matching the robot's current view with the best key-frame in the database. In addition, some other works define other global appearance description methods such as the Discrete Fourier Transform [35], the alternative used by Paya et al. [30] to perform map creation tasks, or Radon Transform [36], used in [37] to find the nearest neighbour in a dense map previously created. Román et al. [38] develop a comparison among these global appearance descriptors performing a mobile robot localization in a real environment under changing lighting conditions [38]. In addition, more recently, deep learning techniques are studied with the idea of creating new global

appearance descriptors. For example, Xu et al. [39] and Leyva et al. [40] proposed holistic descriptors based on a CNN (Convolutional Neural Network) to obtain the most probable robot position and Cebollada et al. [11] carry out a comparison of analytic global-appearance descriptors and CNN-based descriptors in localization tasks.

2.2. Mapping and Clustering Methods

Another important research topic in mobile robotics is map creation. In the related literature, two main different frameworks were proposed in order to obtain maps in mobile robotics. First, metric maps, which represent the environment with geometric accuracy. Second, topological maps, which describe the environment as a graph containing a set of locations with the related links among them, with no metric information. Regarding these options, some authors proposed storing information in the map hierarchically, into a set of layers that contain information from the environment with different levels of granularity. Such arrangement permits solving localization efficiently, in two phases. First, a rough, but fast localization is carried out using the high-level layers; second, a fine localization is performed in a smaller area using the low-level layers. Therefore hierarchical maps constitute an efficient alternative to build maps with autonomous mobile robots [41–43].

Focusing on hierarchical mapping, Balaska et al. [44] develop an unsupervised semantic mapping and propose a localization method. SURF points are used to carry out the clustering and the map is corrected by means of odometry. Korrapati and Mezouar [45] perform clustering and propose image loop closure with the objective of building hierarchical maps. They work with omnidirectional images and local features. Kostavelis et al. [4] propose an augmented navigation graph, an extreme hierarchical map that consist of 4 layers. On the lowest layer, metric information is stored, so it is easier for the robot to navigate and perform localization tasks. As the level of the layer increases so the abstraction level does, in such a way that the highest-level layer is a graph with a conceptual representation of detected places. Clusters on that layer are connected with an indicator that represents probability of success to make a transition to the adjacent cluster.

Additionally, loop closure detection constitutes a crucial step when designing a method to create accurate maps, as shown in [46]. In this work, when a new closure is detected, the new image is stored as a combination of the previous images instead of as a new image. In [47], loop closure detection is performed in two phases using the information of a partially built map: In the first phase, global descriptors are used to find loop closure candidates. In the second one, the loop is closed by choosing the best result among the candidates, using local features. As far as loop closure detection is concerned, the difference of our proposal consists of performing this detection taking advantage of the hierarchical structure of the map.

Finally, as stated in the introduction, we propose building hierarchical maps incrementally. Some authors addressed previously this problem by means of incremental clustering. In [12], a method to build topological maps incrementally is proposed, using the SIFT descriptor, and the number of clusters continuously increases while the robot is navigating. The result is a well-separated clustered route, but the number of clusters tends to be relatively high.

According to [12], incremental clustering constitutes a good choice when the application meets the following criteria. First of all, when the data cannot be easily represented in an n-dimensional space, but it is possible to calculate similarity measures among individuals. Additionally, if the data computation has a high cost and approximate results may be accepted, incremental clustering is faster, allowing us to perform on-line tasks. Finally, if the number of clusters is unknown, incremental clustering methods set the necessary number of clusters depending on different thresholds. These conditions are met in our work since a similarity measure between global-appearance descriptors can be calculated, the method employed does not need to calculate an affinity matrix, easing online operation and the most suitable number of clusters is not known beforehand. Our proposal makes use of omnidirectional images and global-appearance descriptors. Specifically, we used HOG and gist descriptors.

3. Review of Global Appearance Descriptors

In this section, some alternatives to extract global information from panoramic images are summarized. They are known as global-appearance descriptors and they try to keep relevant data with low memory requirements. The visual sensor used in the experiments takes omnidirectional images, for that reason the first step is to transform them into panoramic ones. The starting point is a panoramic image $i(x,y) \in \mathbb{R}^{N_x \times N_y}$ and after using any of the global appearance methods the result is a descriptor $\vec{d} \in \mathbb{R}^{t \times 1}$ where t is the size of the descriptor, as detailed in the next subsections.

Firstly it is necessary to divide the panoramic image in a set of cells. Depending on the shape and number of cells, a different descriptor is obtained. In this work descriptors were built in two different ways. The classic method, described in [48], divides the image in uniformly distributed and non-overlapped horizontal cells. The main idea is that by using panoramic images the descriptor will be invariant to pure rotations of the robot in the ground plane. That is possible due to the fact that the information in the each row is the same and the only change is a horizontal shift in each row. This option was tested in some navigation tasks such as pure visual localization [38], hierarchical localization [3] or topological maps compression [9]. The second method used in this paper consists of defining a set of vertical cells with some overlapping between consecutive cells. To obtain a matching method invariant to the robot orientation, two steps are needed. The descriptor is built by putting together information from the vertical cells, so, the algorithm that compares two descriptors needs, first, to calculate the difference of orientation between both descriptors and shift one of them, by removing its first columns and appending them at the end of the vector. To achieve enough resolution in this step, the descriptor is built with overlapping between consecutive cells. Once the relative orientation is the same, both descriptors can be compared in a straightforward way. Using this additional step that normalizes the orientation, this method also becomes invariant to robot orientation. A comparison between these two methods to build global-appearance descriptors is introduced in [49]. Figure 1 shows how the cells are defined in each of the two methods.

Throughout the paper, the descriptors calculated with horizontal cells are named position descriptors, whereas the descriptors calculated with vertical cells are named orientation descriptors. The objective of using both approaches to build descriptors is twofold. On the one hand, information obtained by descriptors based on pure robot position and by descriptors that are influenced on its orientation are taken into account. On the other hand, this idea could help reduce perceptual aliasing, (i.e., different locations may generate similar visual descriptors). Combining information obtained by horizontal and vertical cells can provide more reliable results as far as image matching is concerned.

These techniques are invariant against changes in the orientation of the robot if it moves in the floor plane and panoramic images are used. To capture them the mobile robot is equipped with an omnidirectional vision system. This system consists of a camera and a hyperbolic mirror mounted in front of the lens. The system is mounted vertically on the mobile robot, as done in several previous works such as [50–53]. The omnidirectional camera captures images with a field of view of 360° around the robot so they offer complete information from the surroundings of the robot from every capture point. Finally, the omnidirectional images might be transformed into panoramic images. The complete experimental setup is presented in Section 5.

Once the methods to divide the image were explained, different approaches to efficiently and robustly describe each region are presented. The Histogram of Oriented Gradient and a descriptor based on gist are the methods used to describe the cells in this work. Table S1 presents the most relevant parameters used in the description process and Table S2 summarizes the parameters that impact the size of the final descriptor.

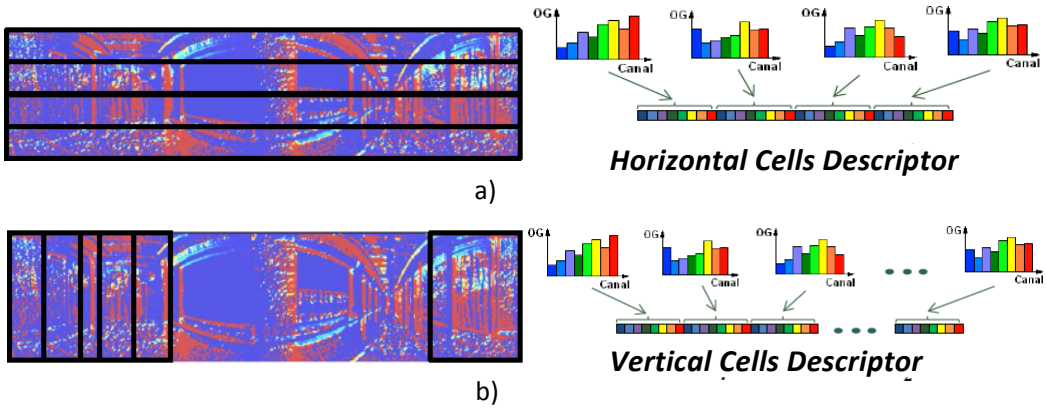


Figure 1. Approaches to build a global-appearance descriptor from a panoramic image: (**a**) with horizontal and (**b**) with overlapped vertical cells.

3.1. Histogram of Oriented Gradient

The Histogram of Oriented Gradient (HOG) was initially used in computer vision to solve object detection tasks. HOG was described by Dalal and Triggs [29], who used it to detect people. This method was later improved in detection and computational cost in the version described in [54]. Afterwards, it was updated by Hofmeister et al. [55], who used a weighted histogram of oriented gradients in small and controlled environments to solve localization from low resolution images. The same authors present a comparison of HOG with other techniques in localization tasks of small robots in reduced environments in [56]. Originally, HOG was built to describe local parts of the scene but it can be redefined to work as a global-appearance descriptor, as in [3], where HOG and other global-appearance descriptors are used to perform hierarchical localization in topological models.

Essentially, it consists of calculating the gradient of the image, obtaining the module and orientation of each pixel. If D_x and D_y represent the derivatives of the image with respect to the x and y axes, respectively, it is possible to calculate the magnitude and orientation of the gradient as:

$$|G| = \sqrt{D_x^2 + D_y^2} \tag{1}$$

$$\theta = \arctan \frac{D_x}{D_y} \tag{2}$$

Afterwards, using a set of cells that covers the whole image it is possible to build the global descriptor using the information of the gradient orientation. To this end, the data is collected in bins, weighting each bin with the module of the gradient of each pixel. Each cell has its own associated histogram and at the end the vector is built concatenating all the histograms. To build the descriptor, the number of bins and cells must be defined. Specifications are shown in Section 6.1.

Classically, this method divided the image into horizontal cells. Authors such as Cebollada et al. [11] or Román et al. [38] used this classical HOG technique to perform localization with a mobile robot. In the present work, as stated before, we will consider both horizontal (with positional intent) and overlapped vertical cells (with orientational intent), as explained at the beginning of the section (Figure 1) . HOG using only horizontal cells is comprehensively described in [48] whereas the second alternative is presented in [49]. Figure 2 shows the method used to build the HOG descriptor, using horizontal cells.

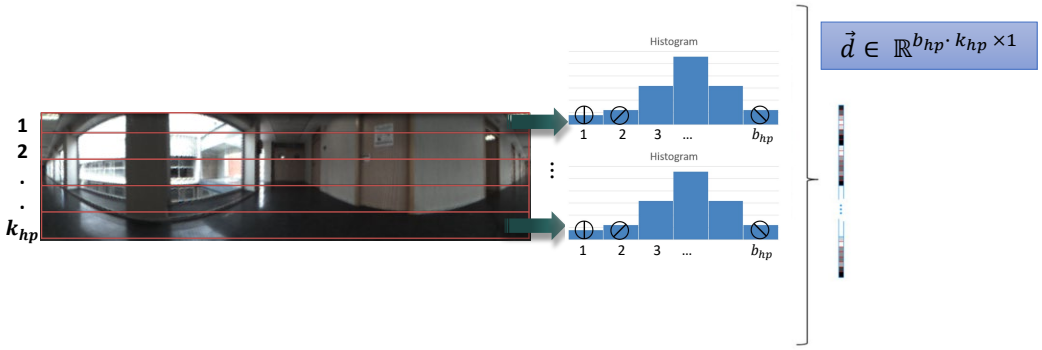


Figure 2. Process to build the HOG descriptor using horizontal cells.

First, regarding the position descriptor, horizontal cells (with the same width as the image) are used, as shown in Figure 1a. Second, in order to obtain the orientation descriptor, vertical cells (with the same height as the image) are used (Figure 1b). Vertical cells are overlapped and separated a distance of $dist_{ho}$ pixels. By shifting the descriptor a rotation of the robot can be simulated.

Once the HOG description is performed, the whole image is reduced to a vector whose size will depend on the number of cells and bins as Figure 1. First, the position descriptor, b_{hp} is the number of bins per histogram and k_{hp} is the number of cells in which the image was divided. The descriptor size is $\vec{d_p} \in \mathbb{R}^{b_{hp} \cdot k_{hp} \times 1}$. Second, in the case of the vertical cells descriptor, the cells are overlapped. This descriptor introduces a new parameter $dist_{ho}$ which refers to distance between consecutive cells. k_{ho} is the number of cells, and it is calculated by $N_y / dist_{ho}$, where N_y is the number of columns of the panoramic image. b_{ho} is still referring to the number of bins per histogram. At the end, HOG with vertical cells reduces a panoramic image into a vector whose size is $\vec{d_o} \in \mathbb{R}^{b_{ho} \cdot k_{ho} \times 1}$. The parameters of the descriptors are summarized in Tables S1 and S2, and the values used in the experiments are specified in Section 6.1.

3.2. Descriptor Based on Gist

The descriptor based on gist was initially introduced in [57] and extended in [58]. This descriptor was further developed in studies such as [33] where it was tested in outdoor environments and as a result, the authors obtain a descriptor whose computational cost is relatively reduced. Some other applications can be found in [59], where a navigation system based on gist is tested; in [60], where gist is calculated in panoramic images and is used to solve localization in urban zones; in [61], where a descriptor based on gist is calculated and dimensionally reduced by using Principal Components Analysis and subsequently used to solve loop closure problems. Finally, Cebollada et al. [11] use such methods to create clustering methods and to perform Visual Place Recognition.

The version of the descriptor used in the present work is built from intensity information, obtained after applying some Gabor filters with different orientations to the image in several resolution levels. To reduce the volume of data each filtered image is divided in a set of cells, and the average intensity of each cell is calculated. As in HOG, classical methods divided the image using horizontal cells [11], whereas new alternatives also tried to use this descriptor using vertical ones [49]. Definitions used in this paper are presented in Table S2 and comprehensively described in [48,49]. Figure 3 shows the process to build the gist descriptor using horizontal cells.

Following that, once the image is filtered with the different masks and scales, the algorithm divides each resulting image into horizontal cells (position descriptor) or into overlapped vertical cells (orientation descriptor), as seen in Figure 1, and the average intensity inside each cell is calculated.

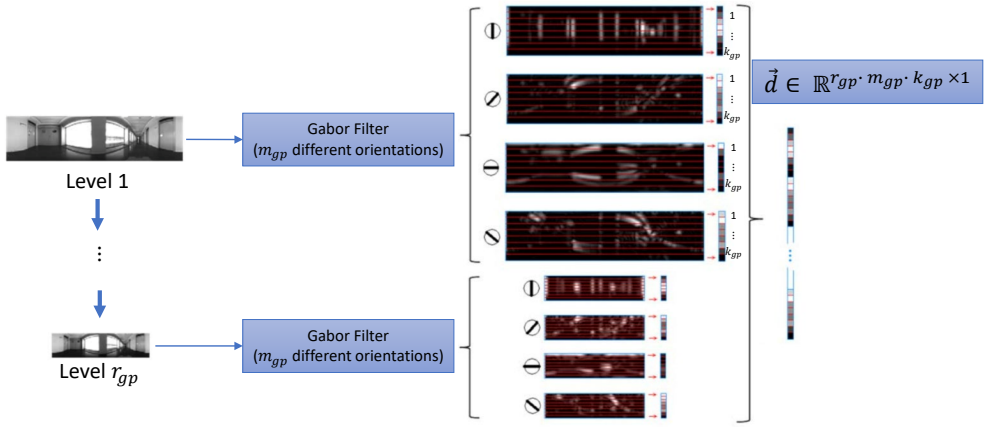


Figure 3. Process to build the gist descriptor using horizontal cells.

In the horizontal-cells descriptor, m_{gp} indicates the number of orientations of Gabor filters, k_{gp} designates the number of cells in which the image was split and r_{gp} indicates the number of different resolution levels used. With these parameters the image can be reduced to a position descriptor whose size is $\vec{d_p} \in \mathbb{R}^{r_{gp} \cdot m_{gp} \cdot k_{gp} \times 1}$. When the cells are vertical, $dist_{go}$ indicates the distance between the beginning of consecutive cells. This parameter is related to the number of vertical cells k_{go} since $k_{go} = N_y / dist_{go}$, where N_y is the number of columns of the panoramic image. m_{go} is the number of orientations of Gabor filters and r_{go} indicates the number of different resolution levels used in orientation descriptor. The orientation descriptor using gist is a vector whose size is $\vec{d_o} \in \mathbb{R}^{r_{go} \cdot m_{go} \cdot k_{go} \times 1}$. The parameters of the descriptors are summarized in Tables S1 and S2, and the values used in the experiments are specified in Section 6.1.

4. Hierarchical Incremental Maps

This section presents the method that we propose to create topological maps incrementally. The starting point is a set of images captured from the same area of the environment, and a node composed of these images. The aim of the process is to build a hierarchical and incremental map, where visually similar zones are detected, compacted and represented as a node. It will be carried out by clustering images with similar features. Broadly speaking, the hierarchical map is built in such a way that when a group of new images do not belong to a previously visited node, a new node is created with them. The hierarchical incremental map is updated and extended every time the robot captures a new image or group of images. A summary of all the parameters needed to follow the process is included in Table S3 and some tunable parameters that can modify and improve the mapping results are shown in Table S4.

When a newly acquired image I_q arrives, firstly, a Node Level Loop Closure is performed with the nodes $N^C = \{N_1, N_2 ..., N_C\}$ currently contained in the map. N^* will represent the set of candidate nodes, so if the node N_i leads to the loop closure, N_i is elected as candidate node and it becomes part of N^*. After retrieving the set of candidate nodes to which the image I_q may belong, secondly, an Image Level Loop Closure is performed with the images that belong to its nodes I^{N^*}.

To carry out these two processes, the position and orientation descriptors are obtained from image I_q. Descriptors should be able to retrieve properly both the candidate nodes and the image that better matches I_q among the images contained in the candidate nodes. At this point, position and orientation descriptors will be only compared with the reference images contained in the nodes which were selected after Node Level Loop Closure $I^{N^*} = \{\cup_{N_i \in N^*} I^{N_i}\}$, where I^{N_i} is the set of images belonging to the node N_i and N^* is the set of nodes selected in the Node Level Loop Closure process. If the Image Level Loop Closure process is successful, a unique image is retrieved as match. If the retrieved image I_i fulfills the Prominence Condition (Section 4.3) and the Centroid Condition Section (4.4), I_q is added to the corresponding node.

Additionally, it is possible that no node is selected in the Node Level Loop Closure process and N^* becomes an empty set ($N^* = \emptyset$). When several consecutive images produce this result ($N^* = \emptyset$) in the Node Level Loop Closure process, a new cluster is created, expanding the hierarchical incremental map. Figure 4 shows a schematic overview of the hierarchical mapping method. The next subsections describe in detail these processes.

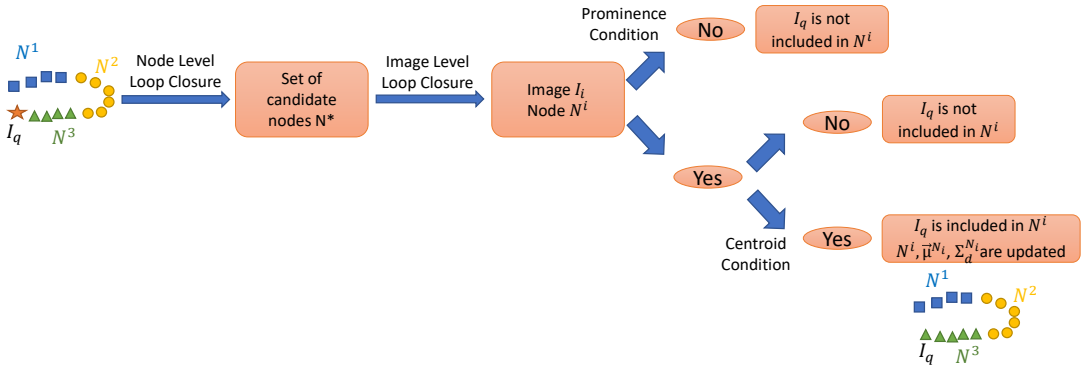


Figure 4. Graphical summary of the proposed method to create hierarchical maps incrementally.

4.1. Node Level Loop Closure

The similarities between each reference node and a new image I_q are evaluated using this method. Nodes are clusters that represent compact zones of the environment, and they contain images with similar features. Each image is described using one of the global appearance descriptors presented in Section 3. Each node N_l is represented through the mean descriptor $\vec{\mu}^{N_l}$ and the covariance matrix $\sum_d^{N_l}$ computed from the descriptors of the images contained in the node.

Similarities between a node and an image are evaluated using the Mahalanobis distance [62]. If $\vec{d}_q$ is the descriptor of the image I_q, the distance $\Delta_{\vec{d}_q}^{N_l}$ between the descriptor $\vec{d}_q$ and the node N_l can be calculated as:

$$\Delta_{\vec{d}_q}^{N_l} = (\vec{d}_q - \vec{\mu}^{N_l})^T (\sum {}_d^{N_l})^{-1} (\vec{d}_q - \vec{\mu}^{N_l}) \quad (3)$$

where i = 1,2,...,C and C is the current number of clusters.

To decide which are the candidate nodes for the closure, the *Mahalanobis distance* has to satisfy the condition of similarity presented in Equation (4), where $\mu_{ns}^{N_l}$ and $\sigma_{ns}^{N_l}$ are the mean and the standard deviation of a Gaussian distribution. When every node is built, 80% of the images are used to model the node, creating with them the mean descriptor $\vec{\mu}^{N_l}$ and the covariance matrix $\sum {}_d^{N_l}$. The other 20% of images are used to build a Gaussian distribution, where $\mu_{ns}^{N_l}$ and $\sigma_{ns}^{N_l}$ represent mean and standard deviation of the distances between each of these 20% of images to the mean descriptor of the node. Also in Equation (4), x is a parameter that must be tuned, and whose value must depend on the number of clusters. The lower x, the more restrictive the condition to create a new node. Values of x used in this work vs. number of clusters can be seen in Figure 5. As shown in this figure, x is less restrictive for a low number of clusters, but it is necessary to limit it from a specific number of clusters. This limit is established depending a parameter (Ω) which has to be tuned (Table S4). x remains constant at {1.7, 1.85, 2, 2.15, 2.3} when $C \geq 9, C \geq 8, C \geq 7, C \geq 6$ or $C \geq 5$ respectively .

$$\left| \Delta_{\vec{d}_q}^{N_l} - \mu_{ns}^{N_l} \right| \leq \left| x \sigma_{ns}^{N_l} \right| \quad (4)$$

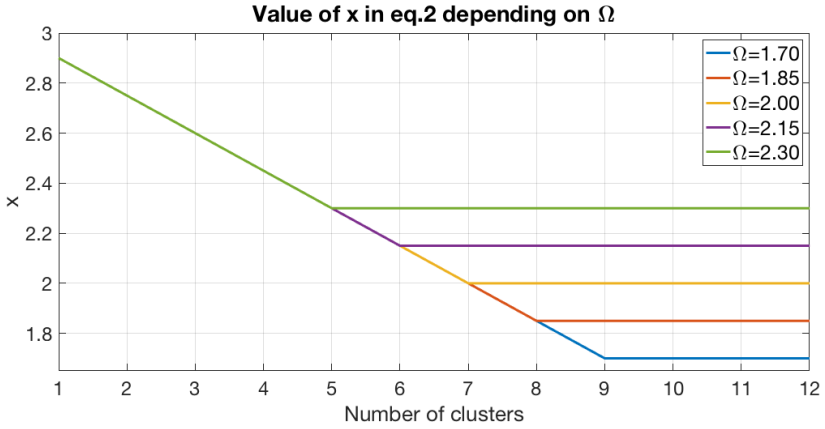


Figure 5. Values of x in the node level loop closure condition versus number of clusters in Equation (4).

At the end of the Node Level Loop Closure, all the nodes that satisfy Equation (4) are considered candidate nodes and introduced in the set N^*. After that, the Image Level Loop Closure is activated to try to retrieve the specific image from these nodes that closes the loop. In some occasions, the Node Level Loop Closure may not find any candidate node that closes the loop. In that case, it outputs an empty set of nodes, and it is considered that the new image does not close the loop with any of the existing nodes so far.

4.2. Image Level Loop Closure. Position and Orientation Descriptors

This algorithm activates if the Node Level Loop Closure is successful and outputs a non-empty set N^* of candidate nodes. In this case, it is necessary to determine which specific image or images close the loop, being the most similar to I_q. Given the set of candidate nodes N^*, all their corresponding images I^{N^*} are evaluated to find the image I_i which is the most similar image to I_q. The problem is solved using two different similarity metrics, described in Section 3. These similarity metrics are computed between the descriptor of the image I_q and every of the descriptors of the I^{N^*} candidate images. Using both metrics, the comparison combines position and orientation information.

First, position information is obtained comparing the position descriptor of the image I_q with the position descriptors of the images I^{N^*} using *Euclidean distance*. Second, the orientation information is used. In this case the method estimates the relative shift between each candidate image I_i and I_q. Then the panoramic image I_i and its associated descriptor are shifted, in such a way that after this shift, both images have the same relative orientation. Then the distance between the orientation descriptors is calculated using *Euclidean distance* (Equation (5)), where $\vec{d}_q$ is the descriptor obtained from the new image I_q and $\vec{d}_k$ is the descriptor of each of the images I_k contained in the candidate nodes N^* after running the Node Level Loop Closure process.

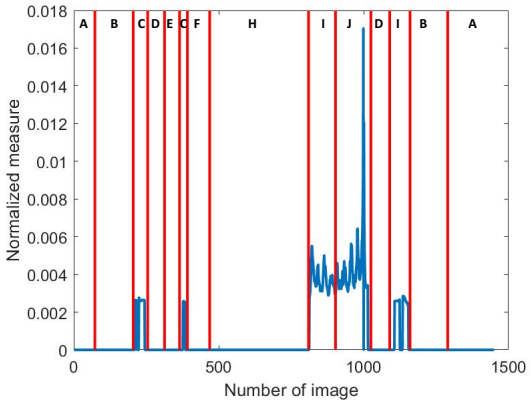
$$dist_{eucl}{}^{\vec{d}_k}_{\vec{d}_q} = \sqrt{\sum_{j=1}^{t} ((\vec{d}_q(j) - \vec{d}_k(j))^2} \tag{5}$$

This process is repeated for all the images I_k contained in the set of candidate nodes N^*. After that, the method calculates the inverse of these distance measures, so the images with small distances (images that are more similar) will have higher similarity measure. Then, the result is normalized in such a way that the sum of the similarities is equal to 1. Next, the position and orientation similarity measures are multiplied to obtain the final similarity measure that combines both kinds of information. The image I_i with the associated higher result is then retrieved by the Image Level Loop Closure (and its node as the most similar node to I_q. It is possible to see an example of these measures in Figure 6, where Figure 6a shows the similarity measures between the image I_q and the images in the candidate nodes when using position descriptors; Figure 6b shows the same similarity comparison but using orientation descriptors and Figure 6c shows the final similarity measures obtained by multiplying

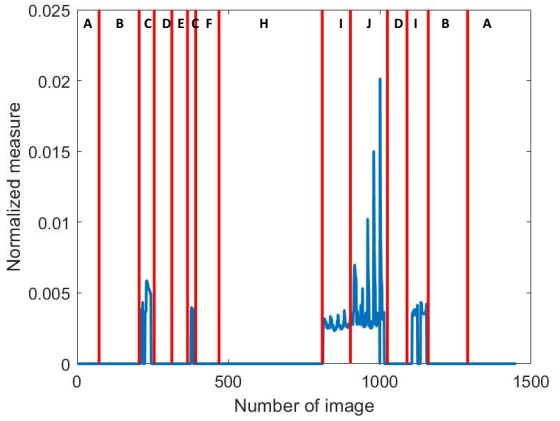
position and orientation measures. These results (final similarity measure) are the data used to retrieve both the image and the node that close the loop with I_q. Only the images I_k that belong to the candidate nodes N^* have a value in these figures, the other images are considered to have null measure.

$$sim^{\vec{d}_k}_{\vec{d}_q} = \frac{1}{dist_{eucl}{}^{\vec{d}_k}_{\vec{d}_q}} \tag{6}$$

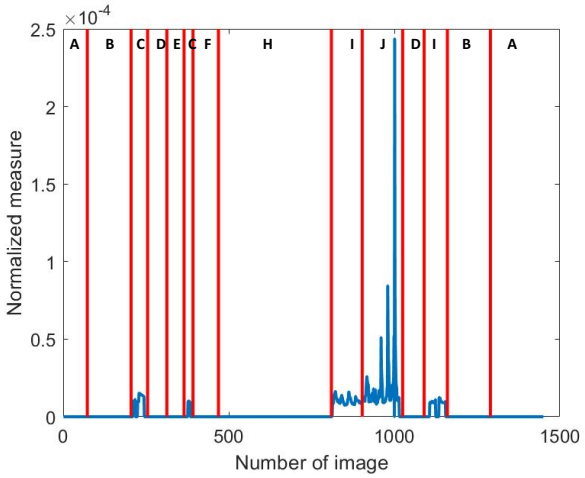
$$i = \underset{k}{\text{argmax}}(sim^{\vec{d}_k}_{\vec{d}_q}) \tag{7}$$



(a) Similarity between position descriptor of I_q and position descriptors of the images in the candidate nodes.



(b) Similarity between orientation descriptor of I_q and orientation descriptors of the images in the candidate nodes.



(c) Final similarity between descriptor of I_q and descriptors of the images in the candidate nodes.

Figure 6. Node Level Loop Closure process example. The node labels are arranged along the top of each subfigure. In this example, the Node Level Loop Closure process has previous retrieved nodes C, I and J.

If no node is selected as candidate node $(N^* = \emptyset)$ the Image Level Loop Closure process cannot be run. Therefore, the image I_q is not assigned to any node, and it remains unclassified, waiting for information from the next images (those which are subsequently captured by the robot). When a set of consecutive images are left unclassified, a new node has to be created with them and the node

representatives are recalculated (the node representative is the mean descriptor of the images contained in the node).

4.3. Prominence Condition

As explained in Section 4.1, the Node Level Loop Closure process selects N^* possible nodes. Among them, the Image Level Loop Closure process finds the most similar image (I_i) which closes the loop. The node to which I_i belongs, is the finally selected node. Prior to selecting this node N_i, the image detected as the most similar to I_i should meet a prominence condition.

The prominence measures how much the peak of the similarity curve (Figure 6c) stands out due to its intrinsic height and its location relative to other peaks. This condition is taken into account because not only a high peak should occur to close the loop, but also that peak should be very distinct compared to its neighbours. The image selected from the Image Level Loop Closure should fulfil the condition presented in Equation (8). In this equation, P_{I_i} is the prominence value of the candidate image, $\mu(P_{I_{k^*}})$ is the mean prominence of the candidate images contained in N^*. γ is a threshold, which is empirically tuned from the experiments as $\gamma = 5$.

$$P_{I_i} \geq \gamma * \mu(P_{I_{k^*}}) \tag{8}$$

4.4. Centroid Condition

As detailed in the previous subsections, the node Level Loop Closure selects first N^* candidate nodes. Among them, the Image Level Loop Closure finds the most similar image which closes the loop and helps select the final node. In order for a selected node to be accepted as the final node, it must fulfil the condition of Equation (9) below. This condition evaluates how the position of the node representative shifts when the new candidate image is added to the selected node. In the condition, the distance between the node representative before (μ^{N_i}) and node representative after adding the descriptor of the new image ($\mu^{N_i \cup \{\vec{d}_q\}}$) has to be lower or equal to the maximum influence on the node representative effected by the images already contained in the node.

$$\left| \vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d}_q\}} \right| \leq \max_j \left(\left| \vec{\mu}^{N_i - \{\vec{d}_j\}} - \vec{\mu}^{N_i} \right| \right) \tag{9}$$

where $\vec{d}_j$ denote all descriptors of the images contained in the node N_i.

The process presented in Sections 4.1–4.4 is summarized in the pseudocode Algorithm 1. Starting from the current set of clusters in the map $N^C = \{N_1, N_2 ..., N_C\}$, their associated data (N_l,$\vec{\mu}^{N_l}$ *and* $\sum_d^{N_l}$ for $l = 1, 2, ..., C$) and the descriptors of each previous image $\vec{d}_k$, it is possible to determine if a new image I_q, whose descriptor is $\vec{d}_q$, has to be assigned to any previous node or not.

Algorithm 1 Pseudocode for Node and Image Level Loop Closure, applying Prominence and Centroid Conditions.

NodeLevelLoopClosure ($\vec{d}_q$)
$N^C = \{N_1, N_2..., N_C\}$ initial set of clusters
$\vec{\mu}^{N_l}, \sum_d^{N_l}$ mean descriptor and covariance matrix of cluster N_l
for `l=1 to number of clusters C` **do**
    $\Delta_{\vec{d}_q}^{N_l} = (\vec{d}_q - \vec{\mu}^{N_l})^T (\sum_d^{N_l})^{-1} (\vec{d}_q - \vec{\mu}^{N_l})$ (Equation (3))
    if $\left|\Delta_{\vec{d}_q}^{N_l} - \mu_{ns}^{N_l}\right| \leq \left|x\sigma_{ns}^{N_l}\right|$ **then** (Equation (4))
        $N^* \leftarrow N_l$;
    end if
end for
end NodeLevelLoopClosure
ImageLevelLoopClosure ($\vec{d}_q$, $\vec{d}_k$, N^*)
for `k= 1 to images in` N^* **do**
    $dist_{eucl}^{\vec{d}_k}{}_{\vec{d}_q} = \sqrt{\sum_{j=1}^{t} ((\vec{d}_q(j) - \vec{d}_k(j))^2}$ (Equation (5));
    find the most similar image I_i using Equation (7);
    if I_i meets $P_{I_i} \geq \gamma * \mu(P_{I_{k^*}})$ **then** (Equation (8), Prominence Condition)
        if I_i meets $\left|\vec{\mu}^{N_i} - \vec{\mu}^{N_i \cup \{\vec{d}_q\}}\right| \leq \max_j(\left|\vec{\mu}^{N_i - \{\vec{d}_j\}} - \vec{\mu}^{N_i}\right|)$ **then** (Equation (9) Centroid Condition)
            I_i and its node N_i close the loop;
            I_q is included in N_i;
            N_i, μ^{N_i} and $\sum_d^{N_i}$ are updated;
        else
            I_i does not meet Equation (9);
            I_q is not included in N_i;
        end if
    else
        I_i does not meet Equation (8);
        I_q is not included in N_i;
    end if
end for
if $N^* = \emptyset$ **then**
    I_q is not assigned to any node;
    The existing nodes are not updated and I_q will be evaluated in subsequent steps;
end if
end ImageLevelLoopClosure

4.5. New Node Creation

The methods presented so far detect the node to which the new image I_q belongs. However, as explained before, the new image may not be assigned to any previous node. That may occur either because the new image is quite different to the existing node representatives or because it does not fulfill either the prominence condition (Section 4.3) or the centroid condition (Section 4.4).

Once a considerable number of images consecutively captured are considered not to belong to any cluster, a new cluster N_q is created. The new cluster is modelled with its mean descriptor $\vec{\mu}^{N_q}$ and covariance matrix $\sum_d^{N_q}$. In order to obtain a new cluster which is significantly different to the ones already existing in the map, the average distance between the representative of the new node and the representative of the other nodes (using Euclidean distance) has to be higher or equal to the average distance among the representatives that already exist. If the new cluster does not fulfill this

condition, this new cluster N_q cannot be elected in further Node Level Loop Closure processes. In this way, new images continue to be evaluated without considering this cluster but if they are still not assigned to any node but they are similar enough to the cluster N_q, they are included to that cluster until the distances from the representative of N_q to other representatives is higher or equal to the average distance among the representatives of the other clusters in the map. This condition tries to guarantee that clusters are different enough among them and that they do not spread excessively along the space of features.

4.6. Node Merging

According to the previous processes, the number of clusters constantly increases. Therefore the resulting number of clusters is sometimes higher than necessary and images that represent adjacent and/or visually similar zones may be included in a unique cluster. The proposed method has the possibility to decrease the number of clusters by merging similar nodes when necessary. That is possible because the method detects when two clusters are similar and the map would be more consistent if they are merged. A new condition is introduced and a newly created node N_q has to exceed a specific dissimilarity threshold to the other clusters. If an existing cluster and a new one are similar, they are merged in a unique cluster. To solve this problem the Mahalanobis distance is used as in Node Level Loop Closure process. Equation (10) shows this condition in the node merging process

$$\Delta_{N_q}^{N_o} = (\vec{\mu}^{N_q} - \vec{\mu}^{N_o})^T (\sum\nolimits_{d}^{N_o})^{-1} (\vec{\mu}^{N_q} - \vec{\mu}^{N_o}) \tag{10}$$

where l = 1,2,. . . ,C and C is the number of clusters before creating N_q. To detect if the node N_q should merge with another node N_l, $\Delta_{N_q}^{N_o}$ is evaluated. This distance has to satisfy the condition of similarity presented in Equation (11), where $\mu_{ns}^{N_o}$ and $\sigma_{ns}^{N_o}$ are the mean and the standard deviation of the Gaussian distribution calculated with the data of node N_o, calculated as detailed in Section 4.1. y is a parameter which has to be empirically tuned (Table S4). The higher it is, the less restrictive the condition is, so the algorithm is more prone to merge similar clusters. Experiments show that this threshold should take a value between 1 and 2. If $y = 1$ the condition is very restrictive so less mergers are done and more clusters will be in the final map, but when $y = 2$ the condition is less restrictive, so more mergers are done and less clusters will be at the end. Experiments to test the values of y are carried out in Section 6.

$$\left|\Delta_{N_q}^{N_o} - \mu_{ns}^{N_o}\right| \leq \left|y\sigma_{ns}^{N_o}\right| \tag{11}$$

5. Image Sets for Experiments

The proposed algorithms are tested with several sets of images captured under real operating conditions. The datasets used are the INNOVA dataset [22] captured by ourselves and the COLD database [53], a third party publicly available dataset which provides robot trajectories in some buildings of the Freiburg and Saarbrücken universities. These three trajectories provide a good choice to perform hierarchical incremental mapping because they represent real environments that experience the typical phenomena that can occur during real operation such as noise, occlusions of the images, changing lighting conditions, movement of people or even some objects or pieces of furniture etc. For that reason these sets of images constitute a challenging scenario to test the robustness of the incremental mapping algorithms.

The different image sequences are recorded by a mobile robot, which is equipped with an omnidirectional camera. The catadioptric vision system is made using a hyperbolic mirror mounted in front of the camera on a portable bracket. The program receives the omnidirectional images and it transforms them into panoramic ones and starts the mapping process, updating the map every time that a new image I_q arrives according to Section 4. Additionally, the robot is equipped with wheel encoders and a laser range scan, which are used to obtain the ground truth, for comparative purposes. However, the proposed method does not use these data and it carries out the incremental mapping

with pure visual information. Figure 7 shows the robot and the camera used to capture the INNOVA dataset. Complete information about the equipment used to capture the COLD dataset can be found in reference [53]. Figure 8 shows some sample images from all the datasets.

Figure 7. Mobile robot and its vision system in INNOVA dataset [22].

(**a**) Image from INNOVA

(**b**) Image from Saarbrücken

(**c**) Image from Freiburg

Figure 8. Panoramic images from each of the datasets.

Table 1 shows some specifications about the 3 trajectories and the number of images taken in each trajectory. Each route contains some loops so they permit testing the Node Level and Image Level Loop Closure processes. Finally, we consider essential to say that these environments are a challenging choice because across the route many walls are made of glass and there are lots of windows so the outdoor weather and lighting conditions could have a negative impact upon the mapping task.

Table 1. Distance covered [m] and number of images used for each of the datasets.

Trajectory Dataset	Number of Images	Distance Covered
Route 1 INNOVA	1450	176.26 m
Route 2 Saarbrücken	1021	56.64 m
Route 3 Freiburg	2778	102.68 m

6. Experiments

In this section, the methods proposed for hierarchical incremental map creation are tested using the image sets introduced in the previous section. Firstly some images are taken to create the first cluster in a supervised method. Once the first cluster is created, the process can start in order to create new clusters, incrementally updating the map every time a new image I_q arrives.

6.1. Parameters to Describe the Images

As explained in Section 3, using global-appearance descriptors, each panoramic image is reduced to a vector $\vec{d} \in \mathbb{R}^{l\times 1}$ whose size depends on the parameters used in this description process. These parameters are outlined in Table S2. In this work, we use $b_{hp} = k_{hp} = 16$ with the HOG position descriptor. In this case, the position vector size is $\vec{d}_h \in \mathbb{R}^{256\times 1}$. For the orientation descriptor $b_{ho} = 16$, $k_{ho} = 256$ and $dist_{ho} = 2$, so the panoramic image is reduced to a vector whose size is $\vec{d}_v \in \mathbb{R}^{4096\times 1}$. In the case of gist, the parameters were kept constant in $m_{gp} = k_{gp} = 16$ and $r_{gp} = 2$ for the position descriptor. Using these parameters the descriptor size is $\vec{d}_p \in \mathbb{R}^{512\times 1}$. In the case of the gist orientation descriptor, it is calculated with $m_{go} = 16$, $k_{go} = 256$, $dist_{go} = 2$ and $r_{go} = 1$ and each image is transformed to a vector $\vec{d}_o \in \mathbb{R}^{4096\times 1}$.

6.2. Parameters to Perform the Loop Closure Processes

When a new image arrives, the first step is to know if it belongs to an existing cluster. As detailed in Section 4, to perform the Node Level Loop Closure, the *Mahalanobis distance* is used, (Equation (3)). After that, it is possible to detect if the image descriptor $\vec{d}_q$ may belong to a specific node N_i, storing all the candidate solutions in the set N^*.

Once the candidate nodes are selected, their images are compared with $\vec{d}_q$ using *Euclidean distance*, Equation (5), where d^{N_k} are all the candidate descriptors from the Node Level Loop Closure. The Image Level Loop Closure detects the most similar image among them, named I_i. After detecting I_i, the image has to fulfill the Prominence and Centroid Conditions (Sections 4.3 and 4.4). Furthermore, the process evaluates if two nodes are similar enough to be merged (Section 4.6).

As described in Section 4, these equations depend on different parameters that have an influence on the results. Equation (4) depends on x. It establishes how restrictive the process is to close the loop in the node level. The lower x is, the easier to close the loop. x must depend on the number of clusters (C) and it is tuned as $x = 3.05 - 0.15 * C$; with limits depending on Ω, Figure 5 shows graphically these values. Another parameter is y, which is used in Equation (11) and it influences the node merging; the lower y, the easier to merge similar nodes. Finally, the parameter γ appears in Equation (8). Once I_i is retrieved, this equation evaluates the prominence of the result to ensure that the retrieved image has a peak on the similarity curve higher enough compared to its the neighbours. The higher γ is, the more prominent the peak must be. During the experiments γ is constant, $\gamma = 5$. These parameters are summarized in Table S4.

6.3. Evaluation

The relative performance of a clustering framework can be evaluated by means of the silhouette. Silhouette evaluates the compactness of each cluster, i.e., the degree of similarity between each descriptor and the other descriptors of the same cluster, comparing it with the descriptors that belong

to the other clusters. The average silhouette (S) of all the entities (images descriptors) is calculated with Equation (12). The higher S is, the more similar each descriptor is to the descriptors in its own cluster and the more different to the descriptors in the other clusters. The maximum value of the silhouette is 1, indicating that the resulting clusters contain well-separated images; the descriptors in each cluster are very similar among them and different to descriptors in the other clusters. By contrast, the minimum value is -1, which means that the resulting clusters do not separate correctly the information. In this work, the silhouette is used to evaluate the compactness of the clusters. Therefore, instead of using the similarity in the feature space, world coordinates are used.

$$S = \frac{\sum_{i=1}^{C} s_i}{C} \tag{12}$$

In Equation (12), C is number of clusters and s_i the average silhouette of the descriptors contained in the cluster i. The silhouette of each descriptor $\vec{d}_j$ is calculated using Equation (13) .

$$s_j = \frac{b_j - a_j}{max(a_j, b_j)} \tag{13}$$

where a_j is the average distance between the capture point of $\vec{d}_j$ and the capture points of the other descriptors in the same cluster and b_j is the average distance between the capture point of $\vec{d}_j$ and the capture points of the other descriptors in the different clusters. The information about the capture points is known because the ground truth of the data set is available, so this information is used to quantify the performance of the mapping algorithm. Notwithstanding, it is worth highlighting that the mapping task is carried out with pure visual information.

In addition, the number of clusters obtained after the process will also be shown in order to evaluate how the parameters have an influence upon this number.

6.4. Results

6.4.1. Influence of the Parameters on the Performance of the Algorithm

Figure 9 shows the results obtained with the HOG descriptor, and Figure 10 with gist for (a) INNOVA, (b) Saarbrücken and (c) Freiburg datasets. In both figures, the first row shows the number of clusters obtained once all the images of each trajectory were processed by the proposed algorithm. The second row presents the average silhouette of the descriptors. These figures show the influence of the parameters y and Ω, which are introduced in Table S4.

First, considering parameter y, as expected, the higher y is the fewer clusters will be in the final map. The effect of this parameter is less significant when gist is used, as shown Figure 10. Second, Ω limits x, the parameter used in Equation (4). If this value is low, more nodes can be selected on the Node Level Loop closure so it can be more difficult to obtain a high number of consecutive images without being in a cluster so the effect of creating new nodes decreases. However, if this value is too high it may lead to some images in a row not being assigned to any node when no new cluster should be created in the Node Level Loop Closure process. As Figures 9 and 10 show, when HOG is used, the parameter y has more influence on the number of clusters than Ω does, yet with the gist descriptor, the latter has greater influence.

The second row of each subfigure in Figures 9 and 10 shows the average silhouette after the mapping process. In general, medium values of Ω offer a higher silhouette but the effect of y is more variable. To better study the results, Table 2 shows the maximum values of silhouette and the number of clusters obtained with the different datasets. It also shows the values of the parameters that lead to that maximum silhouette. It shows how values of Ω around 1.85 offer the best results although when using gist descriptor good silhouettes can be also obtained with higher Ω values. Meanwhile, the results are not substantially dependent on y. As it is possible to observe, better results are obtained using the HOG descriptor.

This behaviour was observed among different experiments and situations, which makes us conclude that HOG is a more suitable option to describe the images with the objective of creating a map incrementally. In addition, the silhouette obtained with the Freiburg dataset is substantially lower than the silhouette obtained with the two other datasets. The reason can be twofold. On the one hand, the route contains a large number of images, and it has some spaces which are visually similar, which challenges the mapping algorithm. On the other hand, the Freiburg environment contains several glass walls, and large and numerous windows, which produce saturations in the images and mixing between the information of adjacent rooms. These phenomena also have a negative impact upon the performance of the method.

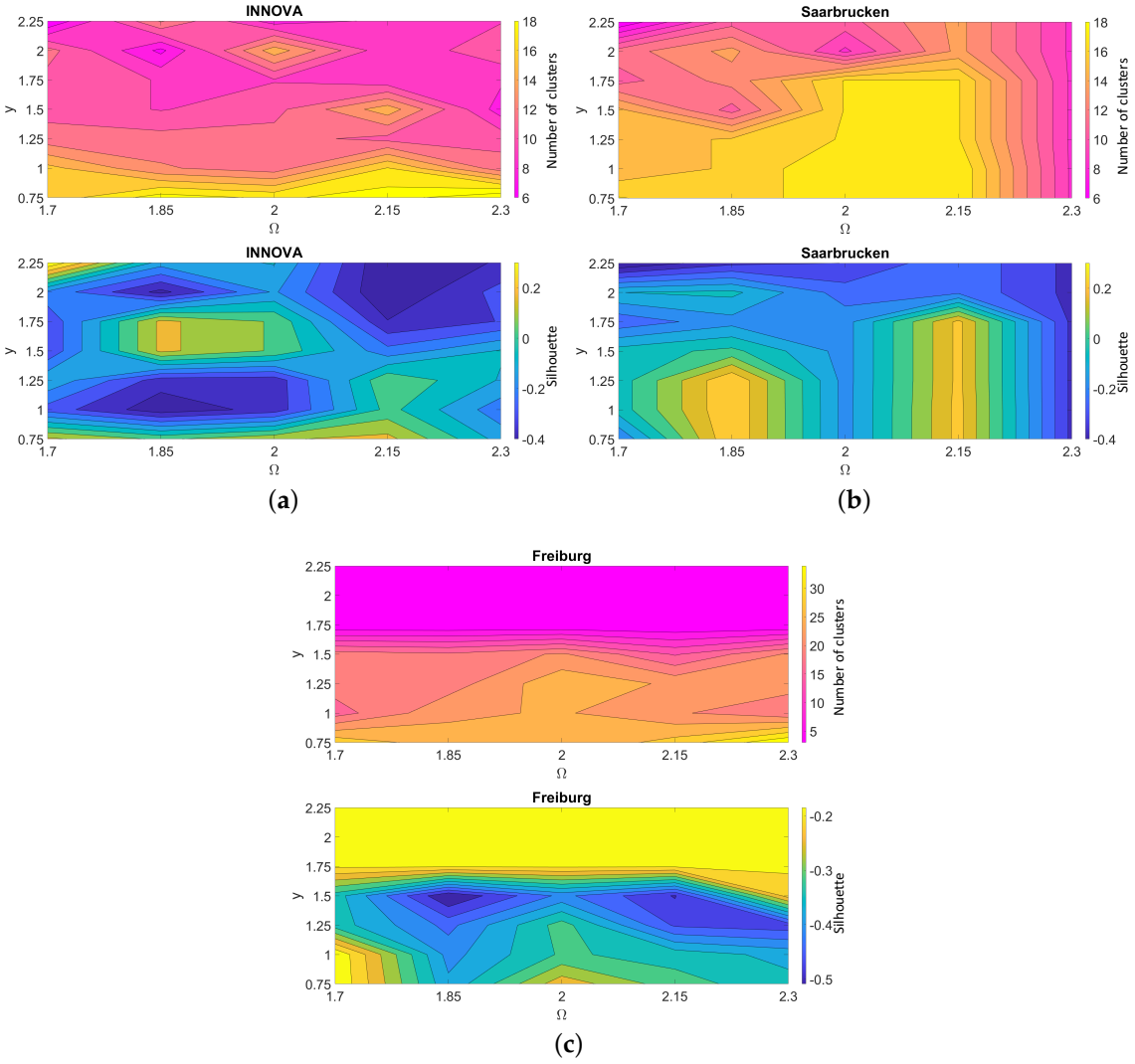


Figure 9. Results obtained with the HOG descriptor for different values of y and Ω for the datasets (**a**) INNOVA, (**b**) Saarbrücken and (**c**) Freiburg. The first row of each subfigure shows the final number of clusters (colour mapped) and the second one the average silhouette (colour mapped) after the proposed incremental hierarchical mapping process.

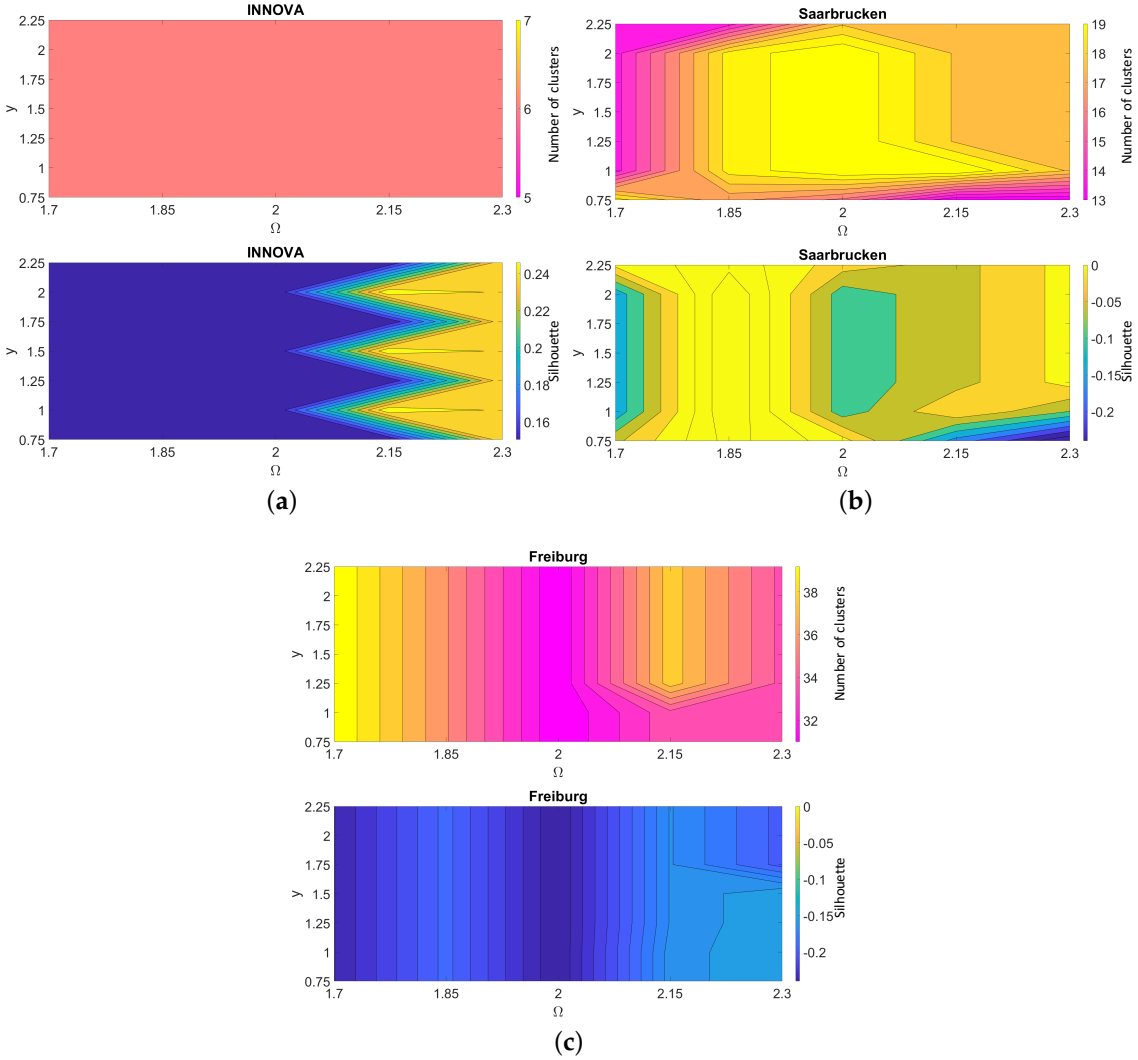


Figure 10. Results obtained with the **gist** descriptor for different values of y and Ω for the datasets (**a**) INNOVA, (**b**) Saarbrücken and (**c**) Freiburg. The first row of each subfigure shows the final **number of clusters** (colour mapped) and the second one the average **silhouette** (colour mapped) after the proposed incremental hierarchical mapping process.

Table 2. Maximum silhouette obtained per configuration, showing also the number of clusters and the configuration of the parameters.

HOG			
Dataset	**Silhouette**	**Number of Clusters**	**Parameters Values**
Route 1 INNOVA	0.3973	6	$y = 2.25, \Omega = 1.7$
Route 2 Saarbrücken	0.2756	16	$y = 1.25, \Omega = 1.85$
Route 3 Freiburg	-0.1526	30	$y = 0.75, \Omega = 1.7$
Gist			
Dataset	**Silhouette**	**Number of Clusters**	**Parameters Values**
Route 1 INNOVA	0.2556	6	$y = 1.5, \Omega = 2.15$
Route 2 Saarbrücken	0.1262	19	$y = 1.5, \Omega = 1.85$
Route 3 Freiburg	-0.1449	34	$y = 1.5, \Omega = 2.3$

6.4.2. Batch Spectral Clustering Results

For comparative purposes, we consider a batch spectral clustering algorithm [11] as benchmark. It is worth highlighting that this batch spectral clustering algorithm has complete information about all the descriptors from the beginning of the process, and can calculate all the mutual similitudes to perform the clustering process. Therefore, it constitutes a powerful benchmark to compare the relative performance of our proposal. To use this batch technique, the number of clusters has to be set initially and all the images must be available, for this reason it is not a good option to create maps incrementally, as the robot explores new areas. Figures 11 and 12 show the results after using a batch spectral clustering method and either HOG or gist descriptors respectively. These figures represent average silhouette versus number of clusters.

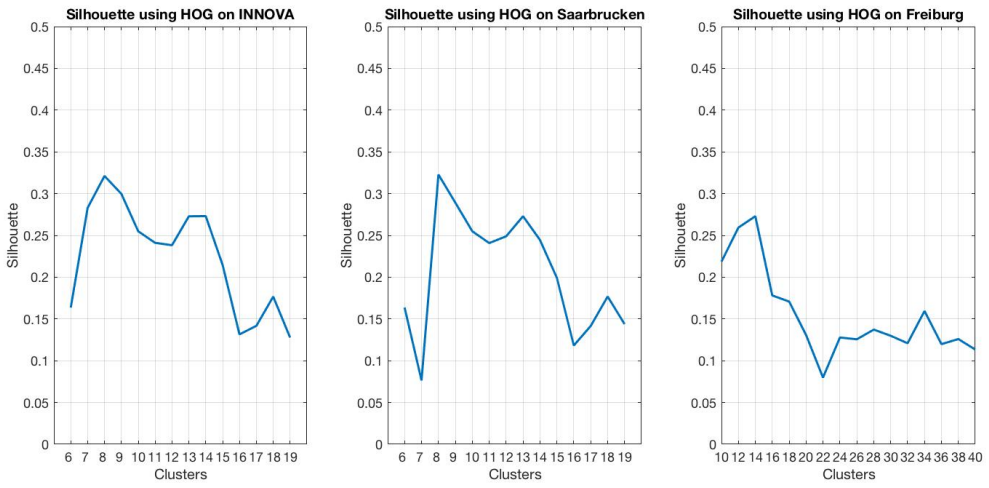


Figure 11. Average silhouette after batch spectral clustering process using HOG versus number of clusters.

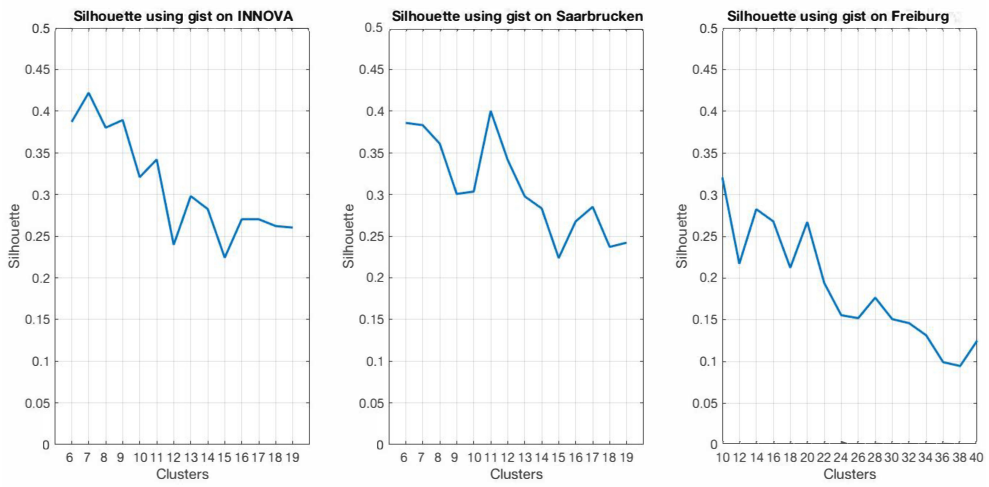


Figure 12. Average silhouette after batch spectral clustering process using gist versus number of clusters.

Figure 11 shows the average silhouette when the HOG descriptor is used. This figure shows that the average silhouette decreases when the number of clusters continuously increases. The results obtained with the INNOVA dataset show that the silhouette is between 0.1 and 0.3; in particular if we observe the result with 6 clusters (which lead to the maximum silhouette with the proposed method) silhouette is 0.13 while with the proposed method is 0.3973. In the case of the Saarbrücken dataset, while the silhouette obtained with batch spectral clustering is between 0.1 and 0.3, the results with proposed incremental method are between -0.4 and 0.3. Observing the result with 16 clusters

(which led to the best silhouette with the proposed method) we note this is about 0.15 as opposed to 0.2756 with the proposed algorithm. Finally, regarding the Freiburg dataset, while the results using batch spectral clustering are between 0.1 and 0.25, results using the incremental method are among −0.5 and −0.15. The maximum value of silhouette of the proposed method is equal to −0.1526 with 30 clusters, while the batch clustering method provides a silhouette of 0.13 with the same number of clusters.

Additionally, Figure 12 show the silhouette when the gist descriptor is used. In this case, the results are less similar to the results obtained with the proposed method. Again, the higher the number of clusters is, the lower the silhouette values are. The best silhouette obtained with gist and the INNOVA dataset is 0.2556 with 6 nodes, the batch clustering method leads to a silhouette equal to 0.38 with the same number of nodes. With the Saarbrücken dataset, silhouette results with a high number of clusters are around 0.25 and 0.4 with the batch method and 0.1262 (19 clusters) with the proposed method. Finally, the results output by the batch clustering method on Freiburg show a silhouette between 0.1 and 0.25 but using the proposed incremental clustering process the maximum silhouette is −0.1449, obtained with 34 clusters. With 34 clusters the silhouette obtained with batch clustering is 0.15. The results of this comparative evaluation between the proposed method and the benchmark algorithm are summarized in Table 3. This table includes the maximum silhouette obtained with the proposed method, and the silhouette obtained with the batch spectral clustering with the same number of clusters. It is necessary to highlight the fact that the proposed method provides better results with HOG in the Innova and Saarbrücken datasets. It is especially relevant, considering that the proposed method permits building the map as the robot explores the environment (i.e., it works with incomplete visual information) but the batch clustering needs to have all the images captured before running the algorithm. The results with the Freiburg dataset are less conclusive, due to the features of this dataset, commented previously. HOG stands out again as an efficient image description method with incremental mapping purposes.

Table 3. Comparison between the maximum silhouette obtained with the proposed method and the silhouette obtained with a batch spectral clustering with the same number of clusters.

HOG			
Dataset	**Number of Clusters**	**Proposed Method**	**Batch Spectral Clustering**
Route 1 INNOVA	6	0.3973	0.13
Route 2 Saarbrücken	16	0.2756	0.15
Route 3 Freiburg	30	−0.1526	0.13
Gist			
Dataset	**Number of Clusters**	**Proposed Method**	**Batch Spectral Clustering**
Route 1 INNOVA	6	0.2556	0.38
Route 2 Saarbrücken	19	0.1262	0.25
Route 3 Freiburg	34	−0.1449	0.15

6.4.3. Bird's Eye View of the Capture Points

Figures 13–15 show a bird's eye view of the capture points of the three datasets in different points of the proposed incremental clustering process. These capture points are shown with different shapes and colors, depending on the cluster they belong to. Figures 13j, 14g and 15j show the result after completing all the proposed incremental mapping method, and the subfigures show how the clusters are being created while the map is updated. Observing the pairs of subfigures: Figure 13c,d,f,g or Figure 15e,f it is possible to see how node merging (Section 4.6) works. Initially, there are several clusters and after including some new images and creating new clusters or making a specific cluster larger, it is possible to obtain similar clusters, so the merging process is launched and it results in a lower number of clusters. Finally, Figure 14b,e shows how transition between rooms resulted in an increased number of clusters. After 399 images, the process has detected 7 clusters but from moment

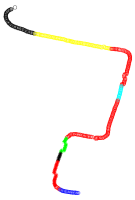
Figure 14e to the end of the process the robot has moved across a long corridor and only changes the room once, for that reason only 2 clusters are created on that process. Others characteristics such as loop closure can be observed in the figures.

(**a**) 240 images and 3 clusters.

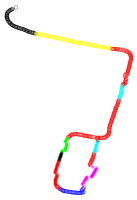
(**b**) 286 images and 5 clusters.

(**c**) 344 images and 5 clusters.

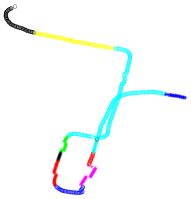
(**d**) 417 images and 4 clusters.



(**e**) 537 images and 8 clusters.



(**f**) 640 images and 10 clusters.



(**g**) 818 images and 11 clusters.

(**h**) 926 images and 12 clusters.

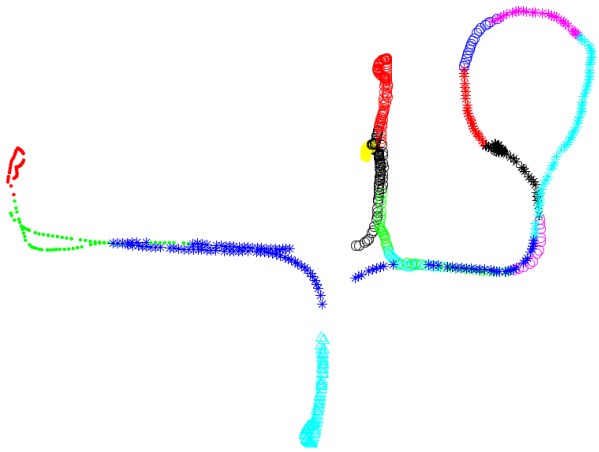
(**i**) 985 images and 6 clusters.

(**j**) 1450 images and 6 clusters.

Figure 13. Maps obtained during the process using HOG $y = 2.25$, $\Omega = 1.7$ with INNOVA dataset. The subfigures show different steps of the process. In the end, the process detects 6 nodes and the final average silhouette is 0.3973. The pairs of **c**,**d**,**f**,**g** show how the node merging process works.

(**a**) 268 images and 3 clusters. (**b**) 399 images and 7 clusters. (**c**) 477 images and 9 clusters.

(**d**) 629 images and 13 clusters. (**e**) 785 images and 14 clusters. (**f**) 886 images and 16 clusters.



(**g**) 1021images and 16 clusters.

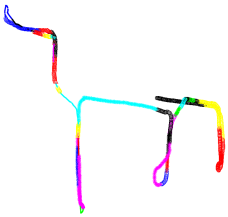
Figure 14. Maps obtained during the process using HOG $y = 1.25$, $\Omega = 1.85$ with the Saarbrücken dataset. The subfigures show different steps of the process. In the end, the process detects 16 nodes and the final average silhouette is 0.2756.

(**a**) 322 images and 3 clusters.

(**b**) 587 images and 9 clusters.

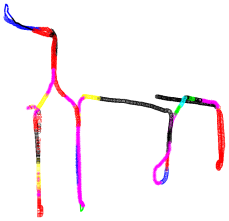
(**c**) 929 images and 13 clusters.

(**d**) 1118 images and 15 clusters.



(**e**) 1581 images and 22 clusters.

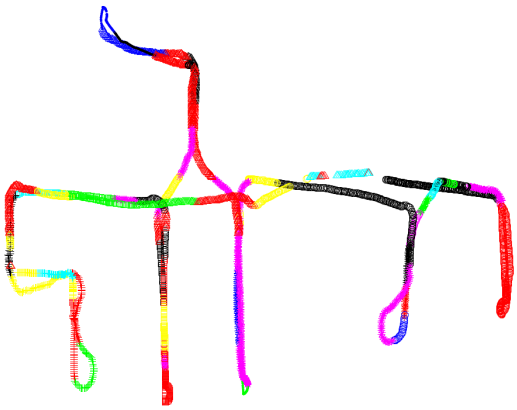
(**f**) 1714 images and 19 clusters.



(**g**) 1957 images and 21 clusters.

(**h**) 2102 images and 24 clusters.

(**i**) 2347 images and 28 clusters.



(**j**) 2778 images and 30 clusters.

Figure 15. Maps obtained during the process using HOG $y = 0.75$, $\Omega = 1.7$ with the Freiburg route. The subfigures show different steps of the process. In the end, the process detects 30 nodes and the final average silhouette is -0.1526. The pair of **e**,**f** shows how the node merging process works.

7. Conclusions

This work presented a method to create hierarchical topological maps incrementally, updating the map every time a new image arrives. The framework is based on the development of an incremental clustering algorithm, presented throughout the paper. The experiments were made in real indoor

environments where the robot navigates under real operation conditions, including illumination variations and changes introduced by human activity. The robot is equipped with an omnidirectional vision system, and the only information used to build the hierarchical map are the images captured by this system. To describe the images two different global-appearance methods were considered: HOG and gist. The experimental section showed the performance of the proposed algorithm and the effect of the most relevant parameters on the final result. Also, a comparative evaluation with a batch spectral clustering algorithm is performed.

The relative accuracy of the method is studied by means of the average silhouette, calculated considering as entities the capture points of every image (ground truth). First, HOG proved to output better results, such that the average silhouette obtained with the proposed incremental method is similar to the silhouette output by the batch spectral clustering method despite the fact that the proposed algorithm only has partial information at each step. In fact, if the parameters are tuned properly if can offer better results than the batch spectral clustering. The results show that in the case of HOG, it is especially important tuning correctly the parameter y, as it has a strong influence on the final number of clusters. Also, the performance of the proposed algorithm degrades when the dataset contains an excessively high number of images, presents complex features (i.e., numerous windows or glass walls) or is prone to visual aliasing.

The results might be considered successful since our incremental algorithm starts with a reduced number of images and updates the map (clusters) every time a new image arrives whereas the batch clustering algorithm has complete information on all the descriptors from the beginning of the process, and can calculate all the mutual similitudes to perform the clustering process. Additionally, gist proved to perform less robustly and the silhouettes obtained with batch spectral clustering are relatively higher. Therefore, the proposed incremental method along with omnidirectional images and the HOG descriptor constitutes the most suitable option to perform incremental mapping.

This work opens the door to new research works on incremental hierarchical map creation using global-appearance description methods in mobile robotics. Once we have tested the robustness and efficiency of the methods in real environments including human activity and presence of changes in the position of some objects, the next step is to improve the algorithm and adapt it to be used in large and outdoor environments. In this sense, we will focus in particular on the problem of abrupt changes of the lighting conditions and changes across seasons, as it is one of the issues that may have a more negative impact upon the visual mapping algorithms.

Supplementary Materials: The following are available online at http://www.mdpi.com/2076-3417/10/18/6480/s1, Table S1: General parameters of the global appearance descriptors, Table S2: Parameters that impact the size of the image descriptors; Table S3: Symbols used in the hierarchical incremental mapping processes, Table S4: Parameters that need to be tuned, Figure S1. Time of the process [s] using HOG descriptor depending on the number of images that it contains and γ parameter, Figure S2. Time of the process [s] using gist descriptor depending on the number of images that it contains and γ parameter.

Author Contributions: L.P. and Ó.R. conceived and designed the experiments; V.R. and S.C. performed the experiments; V.R., L.P. and S.C. analyzed the data; V.R. and L.P. implemented the necessary software. The paper was written and revised collaboratively by all the authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Generalitat Valenciana and the FSE through the grants ACIF/2018/224 and ACIF/2017/146, by the Spanish government through the project DPI 2016-78361-R (AEI/FEDER, UE): "Creación de mapas mediante métodos de apariencia visual para la navegación de robots", and by Generalitat Valenciana through the project AICO/2019/031: "Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales".

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Colleens, T.; Colleens, J. Occupancy grid mapping: An empirical evaluation. In Proceedings of the 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6.
2. Werner, S.; Krieg-Brückner, B.; Herrmann, T. Modelling navigational knowledge by route graphs. In *Spatial Cognition II*; Springer: Berlin, Germany, 2000; pp. 295–316.
3. Cebollada, S.; Payá, L.; Román, V.; Reinoso, O. Hierarchical localization in topological models under varying illumination using holistic visual descriptors. *IEEE Access* **2019**, *7*, 49580–49595. [CrossRef]
4. Kostavelis, I.; Charalampous, K.; Gasteratos, A.; Tsotsos, J.K. Robot navigation via spatial and temporal coherent semantic maps. *Eng. Appl. Artif. Intell.* **2016**, *48*, 173–187. [CrossRef]
5. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
6. Grudic, G.Z.; Mulligan, J. Topological Mapping with Multiple Visual Manifolds. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2005; pp. 185–192.
7. Valgren, C.; Lilienthal, A.J. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robot. Auton. Syst.* **2010**, *58*, 149–156.
8. Štimec, A.; Jogan, M.; Leonardis, A. Unsupervised learning of a hierarchy of topological maps using omnidirectional images. *Int. J. Pattern Recognit. Artif. Intell.* **2008**, *22*, 639–665. [CrossRef]
9. Payá, L.; Mayol, W.; Cebollada, S.; Reinoso, O. Compression of topological models and localization using the global appearance of visual information. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5630–5637.
10. Zivkovic, Z.; Bakker, B.; Krose, B. Hierarchical map building and planning based on graph partitioning. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, 15–19 May 2006; pp. 803–809.
11. Cebollada, S.; Payá, L.; Mayol, W.; Reinoso, O. Evaluation of clustering methods in compression of topological models and visual place recognition using global appearance descriptors. *Appl. Sci.* **2019**, *9*, 377. [CrossRef]
12. Valgren, C.; Duckett, T.; Lilienthal, A. Incremental spectral clustering and its application to topological mapping. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4283–4288.
13. Cha, Y.; Kim, D. *Omni-Directional Image Matching for Homing Navigation Based on Optical Flow Algorithm*; IEEE: Jeju Island, Korea, 2012;
14. Hata, A.; Wolf, D. *Outdoor Mapping Using Mobile Robots and Laser Range Finders*; IEEE: Cuernavaca, Mexico, 2009; pp. 209–214. [CrossRef]
15. Neto, L.B.; Grijalva, F.; Maike, V.R.M.L.; Martini, L.C.; Florencio, D.; Baranauskas, M.C.C.; Rocha, A.; Goldenstein, S. A Kinect-based wearable face recognition system to aid visually impaired users. *IEEE Trans. Hum.-Mach. Syst.* **2016**, *47*, 52–64. [CrossRef]
16. Häne, C.; Heng, L.; Lee, G.H.; Fraundorfer, F.; Furgale, P.; Sattler, T.; Pollefeys, M. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image Vis. Comput.* **2017**, *68*, 14–27. [CrossRef]
17. Choi, J.; Ahn, S.; Choi, M.; Chung, W.K. Metric SLAM in home environment with visual objects and sonar features. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4048–4053.
18. Bonin-Font, F.; Ortiz, A.; Oliver, G. Visual navigation for mobile robots: A survey. *J. Intell. Robot. Syst.* **2008**, *53*, 263. [CrossRef]
19. Gálvez-López, D.; Salas, M.; Tardós, J.D.; Montiel, J. Real-time monocular object slam. *Robot. Auton. Syst.* **2016**, *75*, 435–449. [CrossRef]
20. Kriegman, D.J.; Triendl, E.; Binford, T.O. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. Robot. Autom.* **1989**, *5*, 792–803. [CrossRef]
21. Sturm, P.; Ramalingam, S.; Tardif, J.P.; Gasparini, S.; Barreto, J. Camera models and fundamental concepts used in geometric computer vision. *Found. Trends® Comput. Graph. Vis.* **2011**, *6*, 1–183.
22. Amorós, F.; Payá, L.; Marín, J.M.; Reinoso, O. Trajectory estimation and optimization through loop closure detection, using omnidirectional imaging and global-appearance descriptors. *Expert Syst. Appl.* **2018**, *102*, 273–290. [CrossRef]

23. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
24. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
25. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2010; pp. 778–792.
26. Angeli, A.; Doncieux, S.; Meyer, J.A.; Filliat, D. Visual topological SLAM and global localization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 12–17 May 2009; pp. 4300–4305.
27. Murillo, A.C.; Guerrero, J.J.; Sagues, C. Surf features for efficient robot localization with omnidirectional images. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3901–3907.
28. Gil, A.; Mozos, O.M.; Ballesta, M.; Reinoso, O. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Mach. Vis. Appl.* **2010**, *21*, 905–920. [CrossRef]
29. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
30. Payá, L.; Reinoso, O.; Berenguer, Y.; Úbeda, D. Using omnidirectional vision to create a model of the environment: A comparative evaluation of global-appearance descriptors. *J. Sens.* **2016**, *2016*, 1209507. [CrossRef]
31. Payá, L.; Peidró, A.; Amorós, F.; Valiente, D.; Reinoso, O. Modeling environments hierarchically with omnidirectional imaging and global-appearance descriptors. *Remote Sens.* **2018**, *10*, 522. [CrossRef]
32. Oliva, A.; Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [CrossRef]
33. Siagian, C.; Itti, L. Biologically inspired mobile robot vision localization. *IEEE Trans. Robot.* **2009**, *25*, 861–873. [CrossRef]
34. Zhou, X.; Su, Z.; Huang, D.; Zhang, H.; Cheng, T.; Wu, J. Robust Global Localization by Using Global Visual Features and Range Finders Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 218–223.
35. Menegatti, E.; Maeda, T.; Ishiguro, H. Image-based memory for robot navigation using properties of omnidirectional images. *Robot. Auton. Syst.* **2004**, *47*, 251–267. [CrossRef]
36. Radon, J. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Class. Pap. Mod. Diagn. Radiol.* **2005**, *5*, 21.
37. Berenguer, Y.; Payá, L.; Valiente, D.; Peidró, A.; Reinoso, O. Relative Altitude Estimation Using Omnidirectional Imaging and Holistic Descriptors. *Remote Sens.* **2019**, *11*, 323. [CrossRef]
38. Román, V.; Payá, L.; Reinoso, Ó. Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, Porto, Portugal, 29–31 July 2018; pp. 258–265.
39. Xu, S.; Chou, W.; Dong, H. A Robust Indoor Localization System Integrating Visual Localization Aided by CNN-Based Image Retrieval with Monte Carlo Localization. *Sensors* **2019**, *19*, 249. [CrossRef] [PubMed]
40. Leyva-Vallina, M.; Strisciuglio, N.; Lopez-Antequera, M.; Tylecek, R.; Blaich, M.; Petkov, N. TB-Places: A Data Set for Visual Place Recognition in Garden Environments. *IEEE Access* **2019**, *7*, 52277–52287. [CrossRef]
41. Pantazi, X.E.; Tamouridou, A.A.; Alexandridis, T.; Lagopodi, A.L.; Kashefi, J.; Moshou, D. Evaluation of hierarchical self-organising maps for weed mapping using UAS multispectral imagery. *Comput. Electron. Agric.* **2017**, *139*, 224–230. [CrossRef]
42. Hagiwara, Y.; Inoue, M.; Kobayashi, H.; Taniguchi, T. Hierarchical spatial concept formation based on multimodal information for human support robots. *Front. Neurorobot.* **2018**, *12*, 11. [CrossRef]
43. Hwang, Y.; Choi, B.S. Hierarchical System Mapping for Large-Scale Fault-Tolerant Quantum Computing. *arXiv* **2018**, arXiv:1809.07998.
44. Balaska, V.; Bampis, L.; Boudourides, M.; Gasteratos, A. Unsupervised semantic clustering and localization for mobile robotics tasks. *Robot. Auton. Syst.* **2020**, *131*, 103567. [CrossRef]
45. Korrapati, H.; Mezouar, Y. Multi-resolution map building and loop closure with omnidirectional images. *Auton. Robot.* **2017**, *41*, 967–987. [CrossRef]

46. Latif, Y.; Huang, G.; Leonard, J.; Neira, J. Sparse optimization for robust and efficient loop closing. *Robot. Auton. Syst.* **2017**, *93*, 13–26. [CrossRef]
47. Carrasco, P.L.N.; Bonin-Font, F.; Oliver-Codina, G. Global image signature for visual loop-closure detection. *Auton. Robot.* **2016**, *40*, 1403–1417.
48. 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. [CrossRef] [PubMed]
49. Román, V.; Payá, L.; Flores, M.; Cebollada, S.; Reinoso, Ó. Performance of New Global Appearance Description Methods in Localization of Mobile Robots. In *Iberian Robotics Conference*; Springer: Porto, Portugal, 2019; pp. 351–363.
50. Berenguer, Y.; Payá, L.; Ballesta, M.; Reinoso, O. Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors* **2015**, *15*, 26368–26395. [CrossRef] [PubMed]
51. Valiente, D.; Gil, A.; Reinoso, Ó.; Juliá, M.; Holloway, M. Improved omnidirectional odometry for a view-based mapping approach. *Sensors* **2017**, *17*, 325. [CrossRef]
52. Saito, M.; Kitaguchi, K. Appearance based robot localization using regression models. *IFAC Proc. Vol.* **2006**, *39*, 584–589. [CrossRef]
53. Pronobis, A.; Caputo, B. COLD: COsy Localization Database. *Int. J. Robot. Res. (IJRR)* **2009**, *28*, 588–594. [CrossRef]
54. Zhu, Q.; Yeh, M.C.; Cheng, K.T.; Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1491–1498.
55. Hofmeister, M.; Liebsch, M.; Zell, A. Visual self-localization for small mobile robots with weighted gradient orientation histograms. In Proceedings of the 40th International Symposium on Robotics (ISR), Barcelona, Spain, 10–13 March 2009; pp. 87–91.
56. Hofmeister, M.; Vorst, P.; Zell, A. A comparison of efficient global image features for localizing small mobile robots. In Proceedings of the ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), Munich, Germany, 7–9 June 2010; pp. 1–8.
57. Torralba, A. Contextual priming for object detection. *Int. J. Comput. Vis.* **2003**, *53*, 169–191. [CrossRef]
58. Oliva, A.; Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Prog. Brain Res.* **2006**, *155*, 23–36.
59. Chang, C.K.; Siagian, C.; Itti, L. Mobile robot vision navigation & localization using gist and saliency. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4147–4154.
60. Murillo, A.C.; Singh, G.; Kosecká, J.; Guerrero, J.J. Localization in urban environments using a panoramic gist descriptor. *IEEE Trans. Robot.* **2012**, *29*, 146–160. [CrossRef]
61. Liu, Y.; Zhang, H. Visual loop closure detection with a compact image descriptor. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 1051–1056.
62. Mahalanobis, P.C. *On the Generalized Distance in Statistics*; National Institute of Science of India: Jatani, India, 1936.