Robotica (2016) volume 34, pp. 2400–2413. © Cambridge University Press 2015
doi:10.1017/S0263574715000053

Online topological segmentation of visual sequences using the algebraic connectivity of graphs

Jaime Boal* and Álvaro Sánchez-Miralles

Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, Madrid, Spain

(Accepted January 13, 2015. First published online: February 24, 2015)

SUMMARY

In the context of topological mapping, the automatic segmentation of an environment into meaningful and distinct locations is still regarded as an open problem. This paper presents an algorithm to extract places online from image sequences based on the algebraic connectivity of graphs or Fiedler value, which provides an insight into how well connected several consecutive observations are. The main contribution of the proposed method is that it is a theoretically supported alternative to tuning thresholds on similarities, which is a difficult task and environment dependent. It can accommodate any type of feature detector and matching procedure, as it only requires non-negative similarities as input, and is therefore able to deal with descriptors of variable length, to which statistical techniques are difficult to apply. The method has been validated in an office environment using exclusively visual information. Two different types of features, a bag-of-words model built from scale invariant feature transform (SIFT) keypoints, and a more complex fingerprint based on vertical lines, color histograms, and a few Star keypoints, are employed to demonstrate that the method can be applied to both fixed and variable length descriptors with similar results.

KEYWORDS: Computer vision; Mobile robots; Robot localization; SLAM; Topological modeling of robots.

1. Introduction

Topological maps model the environment as a graph, a set of distinct discrete locations or nodes connected by edges that indicate adjacency. They provide a more compact representation of the space that scales well with the size of the environment, and they can be easily converted into navigation commands. A topological mapping or, in general, a topological simultaneous localization and mapping implementation (SLAM) consists of various phases that are depicted in Fig. 1. First, the robot must acquire raw data from its surroundings, using one or several types of sensors (e.g., laser range scanners or cameras), and convert it into meaningful features that describe the location it currently occupies. The features obtained in consecutive sampling steps usually look alike and belong to the same topological location. Hence, the following stage is to extract from the feature sequences distinct regions that will constitute the nodes of the topological map. Every time a node is identified, it has to be compared with the previously visited locations to produce a belief state of where the robot might be. Finally, once the robot's location is certain, either an existing node is updated to reflect its current appearance or a new one is added to the database. This paper concentrates on the node extraction stage, which is a complicated problem that has been tackled in several manners in the literature.

Most of the early solutions involved geometric properties of the environment, and were therefore tied to specific sensing technologies (i.e., range sensors and laser scanners). For instance, Choset and Nagatani[1] employ range sensor measurements to build a generalized Voronoi graph (GVG) whose nodes are assimilated to topological places. In a 2D planar map, like the one they assume, a Voronoi node is a location equidistant to three obstacles, which should correspond to junctions and dead-ends. In practice, under cluttered environments many spurious nodes are identified and the sparsity of the

* Corresponding author. E-mail: jaime.boal@iit.upcomillas.es

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

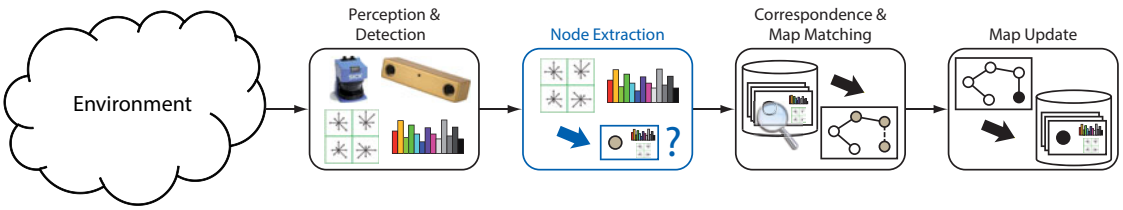


Fig. 1. Different stages in which a topological SLAM system can be divided. This paper deals with the block marked in blue, node extraction.

topological map is lost. Beeson *et al.*[2] overcome this problem through graph pruning, but another issue still persists. If non-structural obstacles are moved, the detected Voronoi vertices change.

Another geometric-based approach uses *gateways* as topological nodes. According to Kortenkamp and Weymouth,[3] a gateway is not only distinctive, but also relevant in the sense that it opens up a new area for the robot to explore. Within this framework, it is worth mentioning the work by Kuipers *et al.*[4] They rely on range sensors to perform a mid-line navigation strategy with a view to finding places where the distance to the lateral obstacles is a local minimum close to a larger maximum. These locations mainly correspond to doorways, and corridor beginning and ending points.

With the incorporation of cameras as a primary source of information, the aforementioned geometric-driven algorithms, which were no longer applicable, were replaced with thresholds on similarity measurements. Tapus and Siegwart[5] use a combination of visual and laser features, coined *fingerprints of places*, to represent a topological location. A new node is then identified whenever the similarity between the last two fingerprints falls below an experimentally defined threshold. Romero and Cazorla[6] follow a similar approach using only computer vision. The descriptor from the current capture is compared to that of the last identified node and, if the similarity is lower than a given threshold, it is subsequently compared with the rest of the nodes in the map using a different threshold. Should there still be no match, a new location is assumed. Although sensor independent, these methods need to be tuned for each specific environment, as an incorrect choice of the threshold parameters can result in too dense or too sparse topological maps.

Rather than using similarity directly, Ranganathan and Dellaert[7] apply the concept of Bayesian surprise, introduced by Itti and Baldi[8] to identify topological locations. With this approach, a place is defined whenever a sudden or unexpected change in the environment occurs. It is based on building a model of the current location to predict the next observation, and looking for a large deviation between this prediction and the actual measurement. The method was successfully tested with both laser range scans and appearance measurements using a bag-of-words model.[9] In a later work, Ranganathan[10,11] uses a Bayesian change-point detection algorithm[12] together with a bag-of-words model as part of a place labeling method.

Another statistically supported alternative is that of Liu and Siegwart,[13,14] who apply a Dirichlet process mixture model to obtain topological regions from two multinomial distributions built from color and geometric features (the distance between vertical lines). Last but not the least, Chapoulie *et al.*[15] employ spherical harmonics extracted from spherical views as descriptors. Using a fixed size sliding window, they build two multivariate normal distributions with which a hypothesis test based on the Neyman–Pearson lemma is performed in order to identify transition points between dissimilar topological locations.

As can be observed, there exist two different approaches to sensor-independent topological location identification. One is based on building a model that allows to apply different statistical algorithms. This path has been proven effective in the literature, but it is not always possible or appropriate. For instance, whenever vision is used, the usual solution is to construct a bag-of-words model. However, for indoor environments it is not easy to find a training set to extract the vocabulary from—unless the same environment is traversed first in order to collect this training set—, and the performance of this method is thus compromised.

A more general approach is to use thresholds on similarity measurements, where no prior model needs to be assumed or built. The benefit of this strategy is that it allows to use more complex matching procedures, even with variable length descriptors; the downside, that thresholds are difficult to adjust correctly. Realize that this method is applicable even to model-based feature representations.

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

For example, Angeli *et al.*[16] evaluate the similarity between two bag-of-words histograms using the Euclidean distance, although any other distance measure can be employed.[17]

This paper proposes a method to extract topological nodes online based on the algebraic connectivity of graphs—which is very closely related to spectral clustering and has been used in robotics for topological segmentation[18]—to address those situations in which only similarities between sensor readings are available without using thresholds directly. However, it is applicable to any type of feature as long as a non-negative similarity measure can be defined.

The rest of the paper is organized as follows. First, Section 2 provides a brief theoretical background on the algebraic connectivity of graphs, followed by the actual topological place identification algorithm proposed. Subsequently, Section 3 presents the results obtained with different types of visual features in an office environment: a bag-of-words model that uses SIFT features, which provides fixed length descriptors that could be treated as multinomial distributions, and VPACK, a combination of color histograms and Star keypoints that result in variable length descriptors depending on the environment, and is therefore not suitable for statistical topological segmentation methods. Finally, the main conclusions are put forward in Section 4.

2. Topological Segmentation Using the Algebraic Connectivity of Graphs

2.1. Theoretical background

The Laplacian matrix, the discrete analogue of the Laplace operator, exhibits a series of interesting properties in the context of graph theory that have been successfully exploited for dimensionality reduction (Laplacian eigenmaps[19]) and clustering applications (spectral clustering[20,21]). One of these properties is related to its second-smallest eigenvalue, known as *algebraic connectivity* or Fiedler value, whose magnitude provides an insight into how well connected a graph is (i.e., the larger the eigenvalue, the better connected it is).[22] The eigenvector associated to this eigenvalue is referred to as Fiedler vector, and can be used to approximate the sparsest cut[23,24] (i.e., partition a graph into two disjoint sets removing as few edges as possible[25]).

In the case of an undirected graph, the Laplacian matrix is built from a symmetric *affinity or adjacency matrix* $A \in \mathbb{R}^{n\times n}$ that encodes the pairwise connectivity between graph nodes and a diagonal *degree matrix* $D \in \mathbb{R}^{n\times n}$ that indicates the number of edges that emanate from any given node. The elements a_{ij} and d_{ij} of these two matrices are computed as shown in Eq. (1).

$$a_{ij} = \begin{cases} 1 \text{ if nodes } i \text{ and } j \text{ are linked} \\ 0 \qquad\quad \text{otherwise} \end{cases} \qquad d_{ij} = \begin{cases} \sum_{j=1}^{n} a_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

There exist multiple definitions of the Laplacian matrix in the literature, each of which has different properties.[23] The most commonly used are the unnormalized Laplacian L, Eq. (2), the symmetric normalized Laplacian L_{sym}, Eq. (3), and the random walk normalized Laplacian L_{rw}, Eq. (4), which owes its name to the fact that it is the transition matrix of the standard random walk. In the equations below, I stands for the identity matrix.

$$L = D - A, \tag{2}$$

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}, \tag{3}$$

$$L_{rw} = D^{-1} L = I - D^{-1} A. \tag{4}$$

A normalized Laplacian is used throughout this chapter as suggested in the work by von Luxburg.[21] This paper explains in detail the properties of the different Laplacian matrices and how they can be applied to perform spectral clustering. The symmetric Laplacian has been chosen, although similar results can be obtained with the random-walk Laplacian. L_{sym} is positive semi-definite, and therefore its eigenvalues λ are always real and greater or equal to zero. In addition, this particular Laplacian matrix verifies that $\lambda \leq 2$. The value of λ_2 for a fully connected graph (i.e., each node is linked to every other node) tends asymptotically to 1 as the number of nodes n grows as shown in Eq. (5).

$$\lambda_{2_{\max}} = \frac{n}{n-1}. \tag{5}$$

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

Algorithm 1: Online Topological Location Segmentation Algorithm

Define : The sliding window size N.
Input : Captures from a camera.
Output: Topological node representative images.

foreach *new capture* **do**
  Form the *affinity matrix* $W \in \mathbb{R}^{n \times n}$, where $w_{ij} \geq 0$ is the similarity between captures i and j for $i \neq j$ and $w_{ii} = 0$.
  Calculate the *degree matrix* $D = \text{diag}\,(d_1, \ldots, d_n)$ with $d_i = \sum_{j=1}^{n} w_{ij}$.
  Obtain the *Laplacian matrix* $L = I - D^{-1/2} W D^{-1/2}$, where I is the identity matrix.
  Compute the second-smallest eigenvalue λ_2 of L.
  if `findValley`$(\lambda_2) > 0$ **then**
    Determine cluster representative (subsection 2.3).
end

As stated by von Luxburg,[21] the concepts and algorithms presented in this section can be extended to any arbitrary type of data if each data point is treated as a node and the connectivity between nodes is replaced by a non-negative similarity measure to indicate the strength of the link.

2.2. Change-point detection algorithm

The problem of recognizing topological places in image sequences is very closely related to clustering, which is one of the applications of the Laplacian matrix as mentioned. The objective is to identify when new images no longer correspond to the scene that was being observed before. At this point, two clusters should be defined. The contributions of this paper are the application of these clustering techniques, which are well-known and widely used in the data mining field (e.g., to classify articles of an online encyclopedia,[26] for speech separation,[27] or in medical image analysis[28]), to topological segmentation in robotics and how to make the algorithm incremental so it can be used online.

In order to identify these change-points, the Fiedler value is used. As the robot moves towards a different topological location, the algebraic connectivity tends to decrease because the latest images start to be dissimilar to the first captures acquired. In terms of graph theory, the new images (nodes) are poorly connected to the initial ones. However, one would expect that the images of the new location form a strongly connected group that makes the algebraic connectivity rise again. This increment can be used as an indicator that the robot has arrived at a different place.

Unfortunately, if the starting location is large compared to the destination, the increment can be marginal and, therefore, difficult to detect. Besides, a large location implies maintaining a big affinity matrix, which involves many pairwise comparison calculations that, depending on the similarity measure, can be computationally demanding. For these reasons, likewise Chapoulie *et al.*,[15] a sliding window that only takes into account the last N images is used. This way, the Fiedler value will clearly increase as more images from the new place enter the window. In order to adjust this parameter, the camera's frame rate and the robot's speed must be taken into account. The lower the frame rate and the higher the speed, the smaller the sliding window should be.

The steps required to perform change-point detection for a new image are put forward in Algorithm 1. To begin with, the affinity matrix is updated with the pairwise similarities between the current capture and the previous $N - 1$ images. As it has to be symmetric, if the similarity measure employed does not exhibit this property, the mean of both pairwise similarities is employed. The affinity matrix is then used to compute the Laplacian matrix, from which the Fiedler value λ_2 is calculated.

If the evolution of λ_2 over time is plotted (Fig. 2), it can be clearly seen that there exist different groups of strongly connected captures. In order to divide this time series into different clusters, several alternatives exist. At the sight of Fig. 2, a possibility would be to look for a significant instantaneous rise in the value of λ_2 or even split it when it reaches zero. However, depending on the type of similarity measure employed, these conditions, especially the second one as will be seen in Section 3, may not be met. Therefore, one of the many algorithms for peak and valley detection is used instead. A pseudocode is provided in Algorithm 2. The approach consists in finding valleys in

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

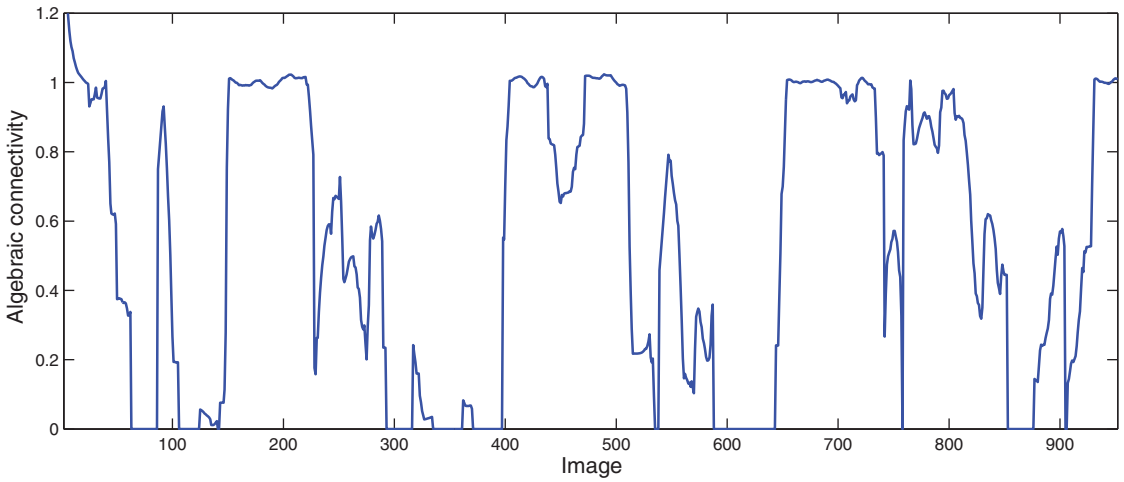


Fig. 2. Sample time series representation of the Fiedler value (λ_2) obtained using the Euclidean distance on a bag-of-words model as similarity measure.

Algorithm 2: Online Valley Detection Function (Adapted from the algorithm by Eli Billauer[29])

Define : $\lambda_{min} \in [0, \lambda_{2_{max}}]$ as the minimum value for a peak to be considered and $\delta \in [0, \alpha]$ as the minimum absolute difference between consecutive peaks and valleys.
Input : The last algebraic connectivity λ_2 computed.
Output: Index of the frame where a valley was detected; 0 otherwise.

```
// Global variables
currentFrameIdx ← 1; // Incremented by the capturing module
lookForMaximum ← true;
maxValue ← −∞;
minValue ← ∞;
minIdx ← 0;
findValley(λ2)
    valleyIdx ← 0;
    // Check if the current algebraic connectivity is an extremum
    if λ2 > maxValue then
        maxValue ← λ2;
    else if λ2 < minValue then
        minValue ← λ2;
        minIdx ← currentFrameIdx;
    end
    // Look for peaks and valleys
    if lookForMaximum then
        if currentValue < maxValue - δ and maxValue ≥ λmin then    // Peak found
            lookForMaximum ← false;
            minValue ← λ2;
    else
        if currentValue > minValue + δ then    // Valley found
            lookForMaximum ← true;
            maxValue ← λ2;
            valleyIdx ← minIdx;
    end
    return valleyIdx;
```

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

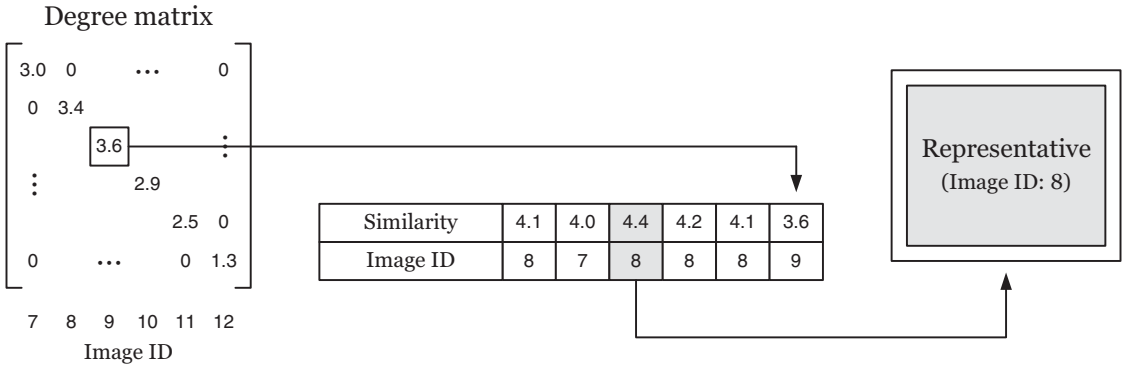


Fig. 3. Node representative selection process. From every degree matrix, the image that is the most similar to the rest of the captures analyzed in the window is stored as a candidate. When a valley is detected, the image with the highest similarity among the candidates that belong to the cluster is chosen as representative.

between two peaks and segment the time series in those points. Two parameters are required to adjust the behavior of the algorithm: α provides a lower bound for peaks (i.e., peaks below this value will be ignored) and δ indicates the minimum difference between consecutive peaks and valleys to declare an extremum. The first parameter allows to ignore image sequences that are not sufficiently strongly connected and can therefore be easily missed in future traversals, whereas the second is used to filter out measurement noise (e.g., a person that briefly passes in front of the robot's camera). α is restricted between 0 and the maximum value of the algebraic connectivity $\lambda_{2_{max}}$ (it tends asymptotically to 1 as N grows), whereas δ ranges between 0 and α. As both thresholds are related, attention should be paid to prevent assigning incompatible values.

2.3. Node representative selection

Once all the captures that form a cluster have been identified, it is necessary to choose a representative image. Conversely to some authors who use the features that correspond to the change-points to describe the topological places,[11,15] we propose using the image that best describes the cluster that is created with the partition (i.e., the frame that is more similar to the rest of the images that form the cluster). This decision is supported by the paper by Stankiewicz and Kalia,[30] which states that whatever is used as a landmark should be persistent, salient, and informative. In other words, it must not disappear with time, it should be easy to detect, and it ought to provide meaningful evidence about the robot's location. However, change-points usually correspond to places with very little information (e.g., turns in a corridor are a common change-point but they all tend to look alike). If topological segmentation is carried out as part of a larger topological mapping or SLAM system, it seems preferable to assign a representative that can be identified with the least possible uncertainty and that cannot be easily missed when revisiting that place even if the frame rate is not high.

Unfortunately, selecting the frame that is most similar to the rest of the members of the cluster involves many pairwise comparisons, especially if the cluster is large. Therefore, in order to keep computational burden under control, this calculation is approximated by taking advantage of the fact that some of these comparisons are already performed to build the affinity matrix (Fig. 3). In every iteration, the sum of the similarities of each frame considered in the affinity matrix is obtained, which are in fact the elements of the main diagonal of the degree matrix. From these, the one with the highest value, along with its total similarity, is set aside. When a representative needs to be determined, it is just a matter of choosing the frame that has the highest similarity value among the stored ones.

3. Results and Discussion

The algorithm presented above, programmed in C++ using OpenCV[31] for image processing and the Armadillo library[32] for linear algebra computations, has been tested with the publicly accessible KTH-IDOL2[33] and COLD[34] image databases that correspond to different office environments.

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

3.1. KTH-IDOL2 database

From the KTH-IDOL2 database, the image sets used were recorded using a manually commanded PowerBot Dumbo robot equipped with a camera installed at a height of 36 cm. The robot was driven at an average speed of approximately 0.3 m/s and the 309 × 240 images were acquired at 5 fps.[35] Two different feature extraction and description methods have been used in order to validate the proposed method. On the one hand, a bag-of-words model constructed from SIFT features[36] matched by means of the Euclidean distance, like the one employed by Angeli *et al.*[16] A 100 word vocabulary was built from the first dataset of KTH-IDOL2 in cloudy illumination conditions, different from the datasets taken at night that were used for testing. These last datasets, which will be referred to as Dumbo night 1 and 2 hereafter, correspond to two different traversals of the same environment following approximately the same path. The former consists of 965 images, whereas the latter has 952 captures. As this method is intended for topological mapping, slight deviations in the trajectory followed by the robot between traversals are not an issue. On the contrary, the fact that the robot is manually driven makes the datasets used for testing more realisitic, because the robot will hardly ever follow exactly the same path.

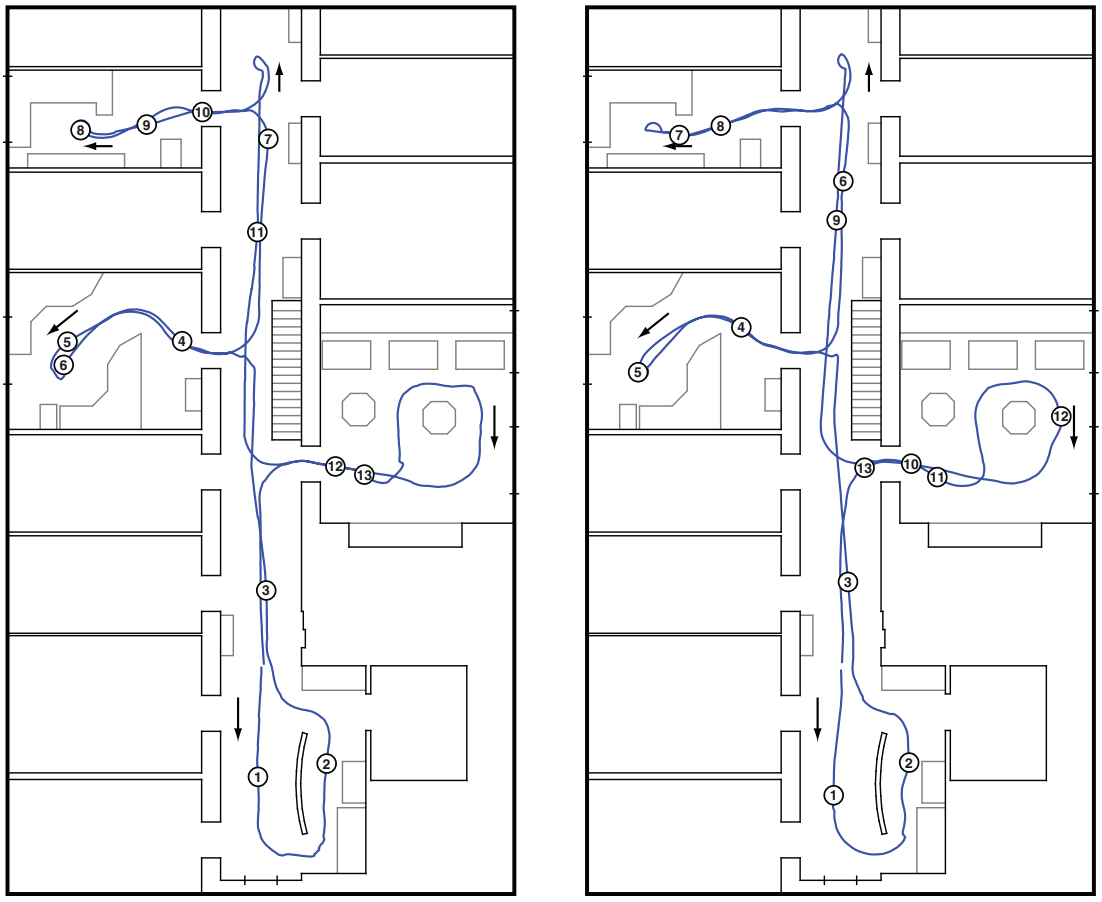
On the other hand, a fingerprint, named VPACK, composed of color histograms and a few Star keypoints extracted from the regions defined by structural vertical lines was used. In this case, similarities are obtained by comparing the histograms and keypoints of both images individually, in pairs, and in trios as if they were letters of a word. This leads to diverse similarity measures that have to be combined somehow. For the tests presented in this paper, the different similarities obtained were combined using the arithmetic mean, although a weighted mean could have also been applied.[37]

These two feature extraction and description methods have been chosen to conduct the experiments because they are qualitatively different. The descriptor of a bag-of-words model is of constant length (the size of the vocabulary) and thus permits to use multivariate statistical techniques. This is one of the reasons why they are employed so extensively. By contrast, VPACK results in variable length descriptors, depending on how many structural vertical lines are identified in the capture, which makes it more difficult to accommodate into a statistical framework. All the results presented below were obtained with a sliding window N of size 25 captures, $\alpha = 0.5$ and $\delta = 0.25$. For this value of N, the maximum possible algebraic connectivity is $\lambda_{2_{max}} = 1.04$. The sliding window size has been adjusted for an average robot speed of 0.3 or 0.35 m/s and a frame rate of 5 fps, which translates into an image being captured every 60 or 70 cm. The length span considered is therefore approximately between 1.5 and 1.75 m, which is reasonable for a normal house or office environment where the visual appearance tends to change rather quickly.

Figure 4 depicts the locations where the representative images of each topological place were captured using the bag-of-words model in the two datasets considered. For completeness, the Fiedler values for Dumbo night 2 with the clusters identified is also provided in Fig. 5. As can be observed, most of the detected topological places are consistent in the two traversals, especially when the robot moves forward for a relatively long time. Nonetheless, occasionally supplementary nodes are introduced (e.g., node 10 in Fig. 4(a) and node 13 in Fig. 4(b)). This may be due to two facts: either the variation in the path followed made a difference in terms of the images acquired or, most probably, the robot moved noticeably slower in that particular area, which resulted in additional similar images being captured. A solution to the latter problem would be to ignore images if the robot is not moving, or increase the sliding window size. Information on the robot's speed could be obtained based on the commands issued, encoders, or simple visual odometry. If this technique were to be employed for a topological SLAM system, any of these extra nodes that is spurious will disappear after several revisits and vice versa, if these nodes are really meaningful, they will end up being detected most of the times. Some of the nodes obtained in both traversals are displayed in Fig. 6. It seems clear that the representatives of corresponding topological places match to a high extent.

Regarding VPACK, the results are also consistent in both traversals as can be observed in Fig. 7. In fact, many of the resulting node representatives are similar to those obtained with the bag-of-words model (Fig. 9). The variations are due to the different nature of the descriptors; the bag-of-words model tends to favor representative images with many keypoints, while VPACK tends to prefer captures with several structural vertical lines. In addition, if one compares the evolution of the algebraic connectivity with both descriptors (Figs. 5 and 8) it can be clearly seen that whereas the Fiedler value reaches zero several times with the bag-of-words model, this never happens with VPACK. The reason behind this behavior is that the wall and furniture colors are similar across the environment and, therefore, that

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press



(a) (b)

Fig. 4. Location of the cluster representatives obtained using the bag-of-words model in Dumbo night 1 (a) and 2 (b). The numbers inside the nodes indicate the order in which they were detected. The robot's path is provided for illustrative purposes only; no odometric information was used while performing this experiment.

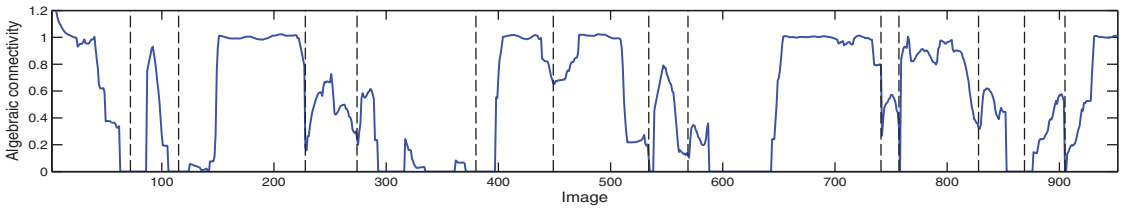


Fig. 5. Fiedler values (λ_2) obtained from the Dumbo night 2 dataset using the bag-of-words model. The dashed lines indicate the division between clusters.

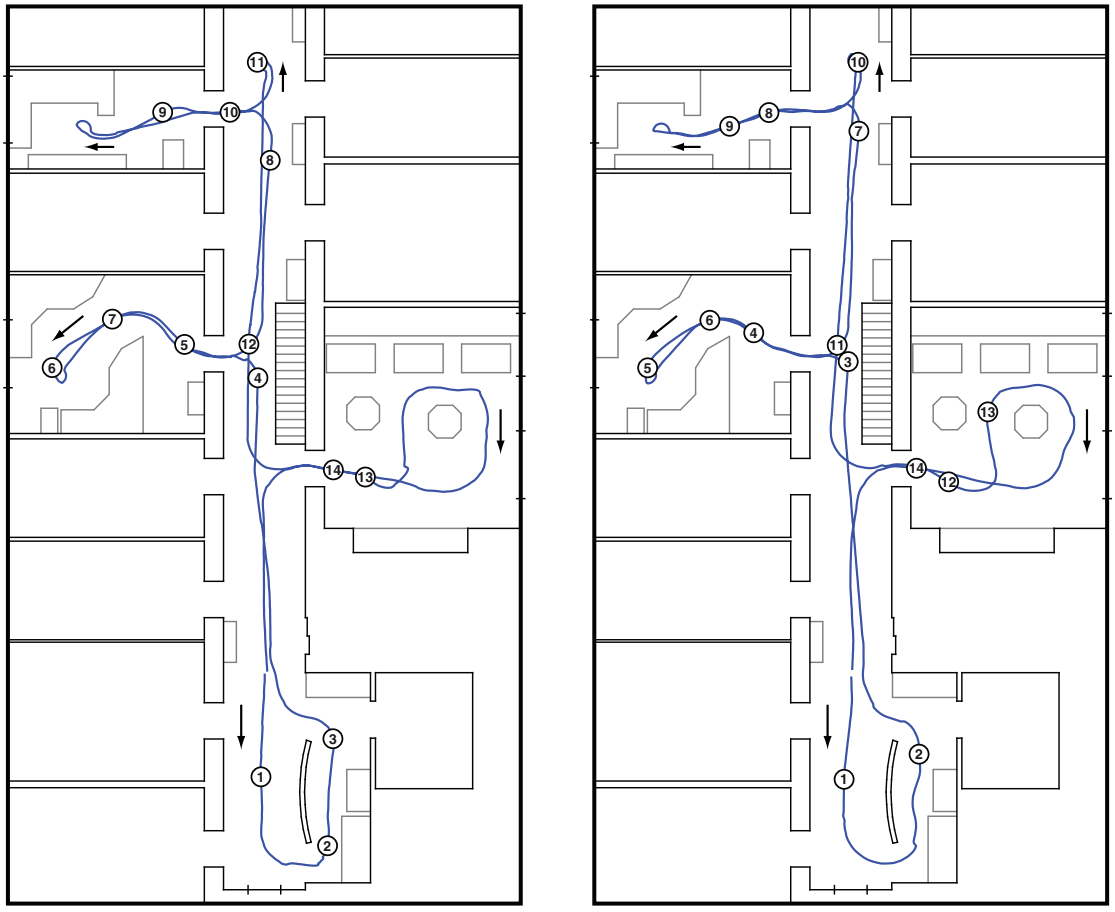
part of the descriptor is never completely different. This is common in an office environment, but should not happen that frequently in a home environment.

In addition, an analysis of the impact of the sliding window size has been carried out. Figure 10 shows the number of nodes identified using VPACK on Dumbo night 1 and Dumbo night 2. The results indicate that, as expected, the amount of nodes extracted reduces as the size of the window grows. The continuous line represents how many of the nodes detected in both traversals match (i.e., have approximately the same representative). With larger window sizes, those areas that are only observed for a limited amount of time are no longer encoded. In the end, representative images correspond mainly to corridors, and room entrances, but no details from inside the rooms are retained, which might not be acceptable for topological localization. Therefore, as aforementioned, it is important to

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press



Fig. 6. Sample node representatives from the same topological locations obtained from Dumbo night 1 (top row) and 2 (bottom row) using the bag-of-words model.



(a) (b)

Fig. 7. Location of the cluster representatives obtained using VPACK in Dumbo night 1 (a) and 2 (b). The robot's path is provided for illustrative purposes only.

correctly estimate the window size depending on the expected number of nodes, the environment, the robot's speed, and the frame rate.

3.2. COLD database

In order to evaluate the performance of the algorithm, another experiment was carried out in the Saarbrücken dataset that is included within the COLD database. The extended version of Path B in a sunny weather light conditions was chosen. The two datasets employed are referred to as Saarbrücken

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

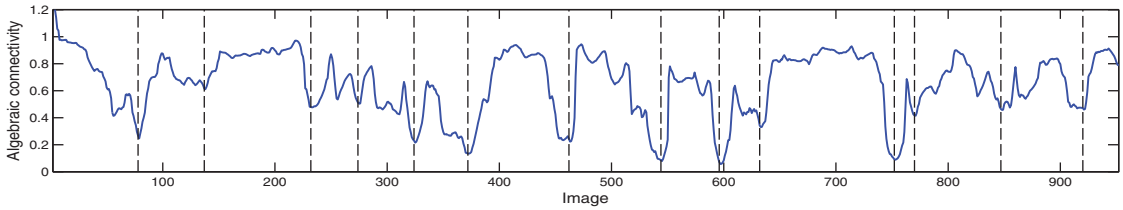


Fig. 8. Clusters identified in the Dumbo night 2 dataset using VPACK features. Dashed lines represent divisions.



Fig. 9. Sample node representatives from corresponding topological places obtained from Dumbo night 1 (top row) and 2 (bottom row) using VPACK.

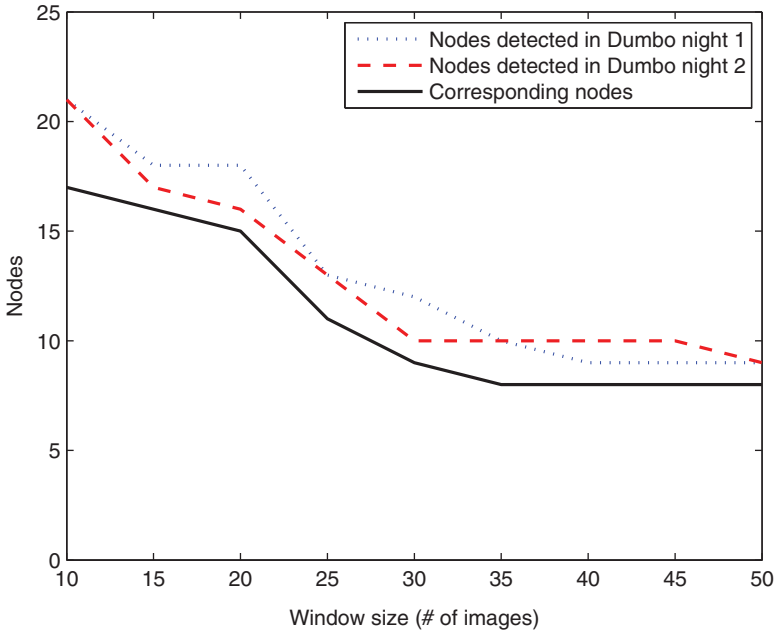


Fig. 10. Evolution of the number of nodes detected using VPACK for different window sizes in Dumbo night 1 and Dumbo night 2 (dotted and dashed lines, respectively). The solid line shows the number of corresponding nodes, that is, those that are identified approximately in the same place in both traversals.

sunny 1 and Saarbrücken sunny 2, and consist of 1104 and 1068 images, respectively, acquired using a perspective camera installed at a height of 140 cm. Note that, contrary to the previous experiment, this camera setup allows to capture images that are more similar to what a human being would see.

In this case, the node representatives obtained in the two traversals using VPACK are almost identical (Fig. 11) except for the fact that in Saarbrücken sunny 1 there is an additional node (node 6). Once again, nodes correspond mainly to entrances, exits, and corridors (Fig. 12). It is also important

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

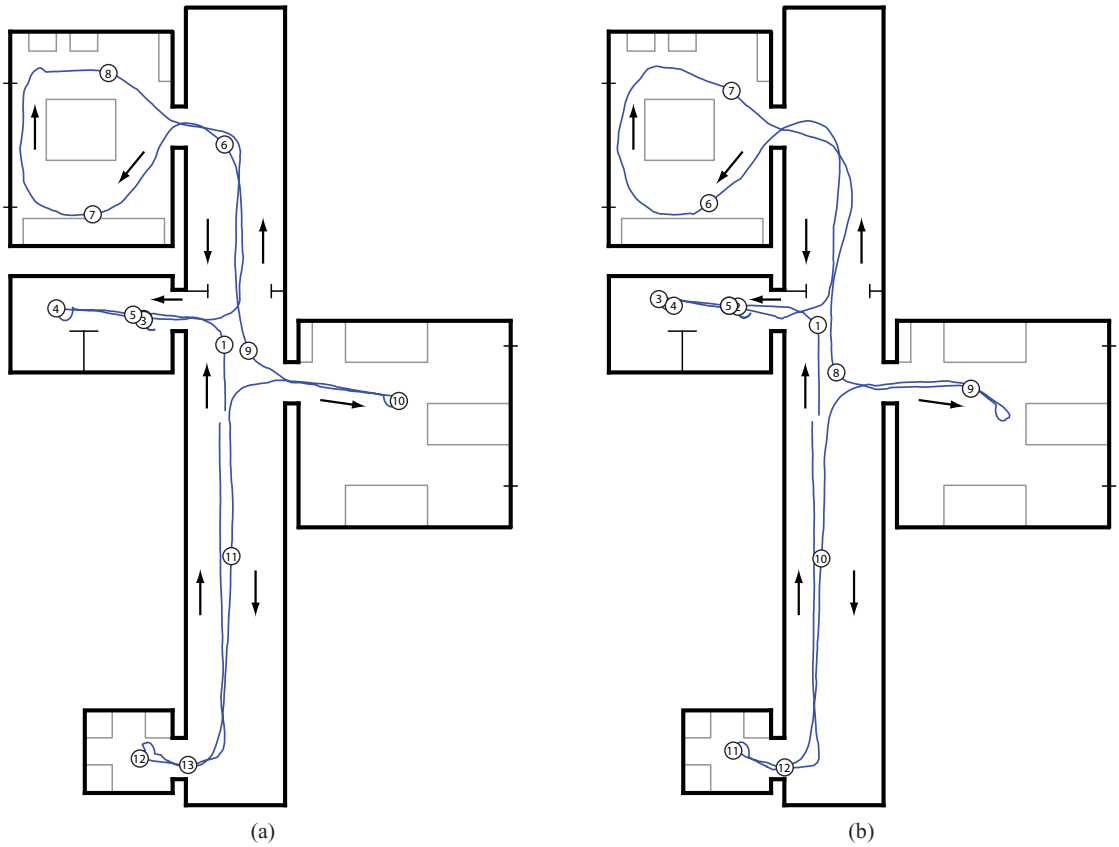


Fig. 11. Cluster representatives obtained using VPACK in Saarbrücken sunny 1 (a) and Saarbrücken sunny 2 (b). Node numbers indicate the detection order. The approximate path followed by the robot is provided as reference.

to note that in the first room the robot enters (the bathroom), four nodes are identified and that some of them overlap. This can be easily explained by the fact that the robot moved back and forth and then stayed still twice. The limitation that was detected in the previous environment also occurs in this experiment: the robot's speed has an impact on the number of nodes detected.

3.3. Discussion

A legitimate question that can be raised is if the starting point or the order in which the rooms are visited has any significant impact on the detected nodes. In the majority of the circumstances it does not because as has been pointed out nodes tend to be identified in room entrances (and exits), which are always gone through with similar orientations. In addition, the proposed algorithm is memoryless. Once it arrives at a new area, the path followed to reach it is no longer taken into account due to the sliding window approach.

However, it is certainly true that if the environment is traversed in the opposite direction some nodes would be extracted in similar places but would have rather different representatives (e.g., corridor nodes). This issue is inherent to directional vision—only omnidirectional cameras can acquire the same image regardless of the orientation—and happens even to human beings. Nonetheless, this is no longer a node detection but a topological SLAM problem and is out of the scope of this paper, which only attempts to provide a robust method to extract topologically meaningful nodes. Based on the similarity of the features of the past and current nodes, the robot would have to consider multiple hypotheses to prevent performing a wrong loop-closure it cannot recover from. However, as enumerating all data association combinations is computationally intractable, a solution would be to employ some sort of sampling method like a particle filter to achieve topological SLAM. A starting point could be the paper by Ranganathan and Dellaert.[38]

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

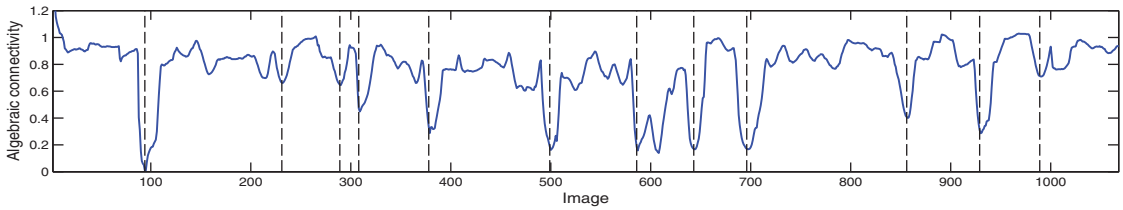


Fig. 12. Clusters divisions found (depicted with dashed lines) in the Saarbrücken sunny 2 dataset using VPACK features.



Fig. 13. Example of node representatives from corresponding topological places obtained from Saarbrücken sunny 1 (top row) and Saarbrücken sunny 2 (bottom row) using VPACK.

4. Conclusion and Future Work

This paper has introduced a method for online topological place extraction in visual sequences based on the algebraic connectivity of graphs, that can be employed for any type of feature detector as long as non-negative pairwise similarities can be defined. Consequently, it is applicable to both constant and variable size feature descriptors. The results of the tests conducted seem to support this assertion. The topological locations obtained are persistent between traversals and could be used to build a complete topological SLAM system. The only step that remains is to apply a mechanism to determine, based on the past observations, whether a newly detected node has already been visited or is a truly new location.

Likewise other similar techniques in the literature, the main drawback of this method is that it assumes a fairly constant speed and, under some circumstances, can output spurious topological nodes when the robot stays still in the same place or moves slower. This is more likely to happen after a close turn in a cluttered room and much less probable during corridor navigation. As aforementioned, this issue can be overcome using odometric information to adjust the size of the sliding window based on the robot's speed. In principle, as there is no need for a precise estimate, direct encoder readings, a simple visual odometry algorithm or even the navigation commands sent to the actuators should be enough.

References

1. H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.* **17**(2), 125–137 (2001).
2. P. Beeson, N. K. Jong and B. Kuipers, "Towards Autonomous Topological Place Detection using the Extended Voronoi Graph," *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 4373–4379.
3. D. Kortenkamp and T. Weymouth, "Topological Mapping for Mobile Robots using a Combination of Sonar and Vision Sensing," *Proceedings of the AAAI 12th National Conference on Artificial Intelligence*, Seattle, WA, USA (1994) pp. 979–984.
4. B. Kuipers, J. Modayil, P. Beeson, M. MacMahon and F. Savelli, "Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy," *Proceedings of the IEEE International Conferemce on Robotics and Automation*, New Orleans, LA, USA, vol. 5 (2004) pp. 4845–4851.

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

5. A. Tapus and R. Siegwart, "Incremental Robot Mapping with Fingerprints of Places," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada (2005) pp. 2429–2434.
6. A. Romero and M. Cazorla, "Topological visual mapping in robotics," *Cogn. Process.* **13**(1 Suppl.), 305–308 (2012).
7. A. Ranganathan and F. Dellaert, "Bayesian Surprise and Landmark Detection," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan (2009) pp. 2017–2023.
8. L. Itti and P. Baldi, "A Principled Approach to Detecting Surprising Events in Video," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 1 (2005) pp. 631–637.
9. J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(4), 591–606 (2009).
10. A. Ranganathan, "PLISS: Detecting and Labeling Places using Online Change-Point Detection," *Proceedings of the Robotics: Science and Systems*, Zaragoza, Spain (2010).
11. A. Ranganathan, "PLISS: Labeling places using online changepoint detection," *Auton. Robots* **32**(4), 351–368 (2012).
12. R. P. Adams and D. J. C. MacKay, "Bayesian online changepoint detection," Technical Report. University of Cambridge, Cambridge, UK (2007). arXiv:0710.3742v1 [stat.ML].
13. M. Liu and R. Siegwart, "DP-FACT: Towards Topological Mapping and Scene Recognition with Color for Omnidirectional Camera," *Proceedings of the IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA (2012) pp. 3503–3508.
14. M. Liu and R. Siegwart, "Topological mapping and scene recognition with lightweight color descriptors for an ominidirectional camera," *IEEE Trans. Robot.* **30**(2), 310–324 (2014).
15. A. Chapoulie, P. Rives and D. Filliat, "Appearance-Based Segmentation of Indoors/Outdoors Sequences of Spherical Views," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan (2013) pp. 1946–1951.
16. A. Angeli, S. Doncieux, J.-A. Meyer and D. Filliat, "Incremental Vision-Based Topological SLAM," *Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems*, Nice, France (2008) pp. 1031–1036.
17. S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.* **1**(4), 300–307 (2007).
18. M. Liu, F. Colas and R. Siegwart, "Regional Topological Segmentation based on Mutual Information Graphs," *Proceedings of the IEEE International Conference Robotics and Automation*, Shanghai, China (2011) pp. 3269–3274.
19. M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.* **15**, 1373–1396 (2003).
20. A. Y. Ng, M. I. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inform. Process. Syst.* **14**, 849–856 (2002).
21. U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.* **17**(4), 395–416 (2007).
22. M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math. J.* **23**(2), 298–305 (1973).
23. F. K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, vol. 92 (American Mathematical Society, Providence, RI, USA, 1997).
24. D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra Appl.* **421**, 284–305 (2006).
25. S. Chawla, "Sparsest Cut," **In**: *Encyclopedia of Algorithms* (M.-Y. Kao, ed.) (Springer, New York, NY, USA, 2008) pp. 868–870.
26. J. Szymański, "Categorization of Wikipedia Articles with Spectral Clustering," **In**: *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, *Lecture Notes in Computer Science* (H. Yin, W. Wang and V. Rayward-Smith, eds.) vol. 6936 (Springer Berlin Heidelberg, Norwich, UK, 2011) pp. 108–115.
27. F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.* **7**, 1963–2001 (2006).
28. T. Schultz and G. L. Kindlmann, "Open-box spectral clustering: Applications to medical image analysis," *IEEE Trans. Vis. Comput. Graph.* **19**(12), 2100–2108 (2013).
29. E. Billauer, "peakdet: Peak detection using MATLAB," (2012). Available at: http://billauer.co.il/peakdet.html, accessed Apr. 27, 2014.
30. B. J. Stankiewicz and A. A. Kalia, "Acquisition of structural versus object landmark knowledge," *J. Exp. Psychol.: Human Perception Perform.* **33**(2), 378–390 (2007).
31. G. Bradski, "The OpenCV library," (2000). Available at: http://www.opencv.org, accessed Apr. 27, 2014.
32. C. Sanderson, "Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments," Technical report, NICTA (2010). Available at: http://arma.sourceforge.net, accessed Apr. 27, 2014.
33. J. Luo, A. Pronobis, B. Caputo and P. Jensfelt, "Incremental Learning for Place Recognition in Dynamic Environments," *Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems*, San Diego, CA, USA (2007) pp. 721–728.
34. A. Pronobis and B. Caputo, "COLD: The cosy localization database," *Int. J. Robot. Res.* **28**, 588–594 (2009).

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press

35. J. Luo, A. Pronobis, B. Caputo and P. Jensfelt, "The KTH-IDOL2 database," Technical Report CVAP304, KTH Royal Institute of Technology, CVAP/CAS, Stockholm, Sweden (2006).
36. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
37. J. Boal, Á. Sánchez-Miralles and M. Alvar, "Matching monocular lightweight features using n-gram techniques for topological location identification," *Robotica* **FirstView**, 1–15 (2014).
38. A. Ranganathan and F. Dellaert, "Online probabilistic topological mapping," *Int. J. Robot. Res.* **30**(6), 755–771 (2011).

https://doi.org/10.1017/S0263574715000053 Published online by Cambridge University Press