Visual Topological Mapping Using an Appearance-Based Location Selection Method

Mohammad Asif Khan[1,2](✉) and Frédéric Labrosse[1](✉)

[1] Department of Computer Science, Aberystwyth University,
Aberystwyth SY23 3DB, UK
{ask2,ffl}@aber.ac.uk
[2] Sukkur IBA University, Airport Road, Sukkur 65200, Pakistan

Abstract. Visual representation of an environment in topological maps is a challenging task since different factors such as variable lighting conditions, viewpoints, mobility of robots, dynamic and featureless appearance, etc., can affect the representation. This paper presents a novel method for appearance-based visual topological mapping using low resolution omni-directional images. The proposed method employs a pixel-by-pixel comparison strategy. Successive images captured as a mobile robot traverses its environment are compared to estimate their dissimilarity from a reference image. Specific locations (nodes in the topological map) are then selected using a variable sampling rate based on changes in the appearance of the environment. Loop-closures are created using a dynamic threshold based on variability of the environment appearance. The method therefore proposes a full SLAM solution to create topological maps. The method was tested on multiple datasets, which were captured under different weather conditions along various trajectories. GPS coordinates were used to stamp each image as ground truth for evaluation and visualisation only. We also compared our method with state of the art feature-based methods.

Keywords: Topological mapping · Appearance-based · Visual SLAM · Loop closure

1 Introduction

Creating and maintaining a map of the environment of a mobile robot is an important aspect of autonomous navigation. Mapping for robots can be done using various sensors such as GPS, magnetic compass, laser scanner, wheel odometry and cameras [5,9,12]. We present a novel pixel-based technique for creating a visual topological map where specific places are represented as the nodes of a graph and navigation links between them as the edges. The selection of places is based on a purely pixel-based image comparison by contrast to more traditional methods that first extract features from the images and then compare them.

A. Mohammad et al. (Eds.): TAROS 2020, LNAI 12228, pp. 90–102, 2020.
https://doi.org/10.1007/978-3-030-63486-5_12

As the robot moves away from a reference location, the dissimilarity between the reference image (captured at the reference location) and the current image increases, creating a catchment area in which the robot can go back to the reference location [7]. We use this property to select a succession of reference locations (nodes) as the robot traverses its environment. Localisation is performed and loop-closures are created as the map is being built using similarity between nodes and thresholds based on local conditions. The paper therefore proposes a full topological SLAM solution.

The paper is arranged as follows. Section 2 discusses related work. Section 3 presents the proposed method in detail while Sect. 4 gives results and compares the proposed method to the state of the art. Finally, Sect. 5 concludes and discusses future work.

2 Related Work

The mapping of large environments can be done using appearance based visual topological maps. Such method has the advantage of being easily scalable compared to occupancy grids and less sensitive to noise than metric maps. Various methods have been proposed in the literature to visually represent the environment of a robot [4]. We review a few of these here.

The concept of probabilistic topological mapping was introduced in [11], where Bayesian inference is used to explore all possible topologies of the map. Measurements (odometry or appearance of the environment) are used in a Markov chain Monte Carlo algorithm to estimate the posterior probability of solutions in the space of all topologies. The appearance used is a Fourier signature of panoramic images. In [14] another Bayesian approach is used that does not require any motion model or metric information but uses the appearance of previously visited places as colour histograms (histograms are often used to reduce the amount of data to be processed, such as in [13]). Both Fourier signatures and colour histograms are poor representations of an environment and location aliasing therefore needs to be explicitly tackled by the methods.

In general, methods using global image descriptors tend to be faster compared to methods that use local descriptors. However, they suffer from problems such as occlusions, illumination effects and aliasing [4].

Local descriptors such as SIFT (Scale Invariant Features Transform, [8]) and variations over them are often used in topological mapping and localisation. These are particularly appropriate to dynamic environments, variable illumination from and varied view points as these descriptor tend to be scale, rotation and illumination independent. The matching of such features for localisation is often used in a Bag of Words (BoW, [1–3]) or Bag of Raw Features (BoRF, [16]) to improve the matching efficiency.

The concept of BoW (Bag of Words) for features matching has been widely used. These require a visual vocabulary of features that can be built online or offline. Most methods build the visual vocabularies offline during a training period, but some methods have been developed for online incremental building.

In particular, in [3] a method called BIN Map has been presented that uses binary features and creates online binary bags of (binary) words. The method is evaluated on different indoor and outdoor environments with good results.

In [1] the Fast Appearance-based Mapping (FAB-MAP 1.0) was proposed; it uses a probabilistic model for recognising places. The probabilities of visual words occurrences are approximated using a Chow Liu tree, offering efficient observation likelihood computation. The observation likelihood was used in a Bayes filter to predict loop closures. The downside of FAB-MAP 1.0 was that with every observation there was a need to compute the likelihood for existing nodes on the map. FAB-MAP 2.0 overcomes this issue [2], making the method scalable to kilometers long trajectories.

Biologically inspired methods have also been proposed to solve the visual mapping problem. A method was proposed which uses an artificial immune system to automatically select images that are representative of a stream of images, ignoring seldom seen images and preserving a regular sampling of the images in image space [10]. Using the concept of ARB (Artificial Recognition Ball) and a NAT (Network Affinity Threshold) similar images were linked while dissimilar images were not. However, the method can produce incomplete topologies when insignificant images are being removed from the map.

3 Appearance Based Visual Topological Mapping

In the visual topological map, different places are represented by nodes of a graph quantified by the panoramic image of the corresponding place. Connection/edges between the nodes correspond to navigable pathways between the places. We describe below how places are selected to be added as nodes of the graph.

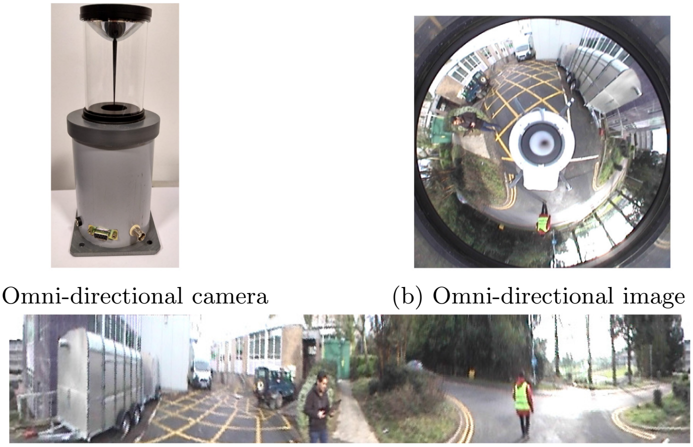
3.1 Image Comparison and Alignment for Mapping

The proposed method uses an omni-directional camera (Fig. 1a). This camera captures omni-directional (360° field of view) images (Fig. 1b), which are then unwrapped into panoramic images (Fig. 1c).

The images are captured at regular time (space) intervals by a mobile robot and each image is stamped with its GPS coordinates obtained using a RTK differential GPS unit.

Image comparison is done using a pixel-wise metric rather than the more usual feature-based method as the holistic metric is more robust to noise and featureless environments [6]. As in other works (e.g. [7]), the comparison between two images I_i and I_j is done using the Euclidean distance where the images are considered as points in a $h \times w \times c$ space, where h and w are the height and width of the images and c is the number of colour components:

$$d\left(I_i, I_j\right) = \sqrt{\sum_{k=1}^{h\times w} \sum_{l=1}^{c} \left(I_i\left(k,l\right) - I_j\left(k,l\right)\right)^2}. \tag{1}$$



(a) Omni-directional camera

(b) Omni-directional image

(c) Unwrapped panoramic image

Fig. 1. Camera and images used for capturing the appearance of the environment

The literature suggests that it is not a good idea to use the RGB colour space from which luminance cannot be removed. Therefore, the CIE L*a*b colour space was used. Luminance was discarded to make the method less sensitive to changes in brightness [15], resulting in using images in the *ab* colour space.

Images are aligned to a common heading so that they can be meaningfully compared. This is done using the Visual Compass presented in [6]. In this method two successive panoramic images are aligned by doing a local optimisation of their similarity as a function of the rotation of the second image. To limit drift the comparison is done relative to a moving reference image selected automatically in the stream of images based on the matching quality of the successive images to that reference image. This is done using the amplitude threshold $\phi = 0.41$ [6].

3.2 Creation of the Topological Map with Adaptive Spatial Sampling

The proposed method automatically selects images from a stream to be nodes of the topological map, the edges corresponding to the traveled path of the robot. Additionally, loops are automatically closed when specific conditions happen.

Contrary to many other methods in the literature (e.g. [2,3]), the spatial sampling is automatically adapted to the environment. This offers variable sampling that ensures that the important events of the environment are being recorded.

Figure 2 shows the Euclidean distance (Eq. (1)) between all the images of a sequence and a reference image at the centre of the sequence (image index 213). This sequence was captured by a robot travelling from a grassy patch to another one separated by a road. It can be seen that the Euclidean distance increases as the robot moves away from the reference location (corresponding to the centre image), creating a catchment area. It has been shown that the

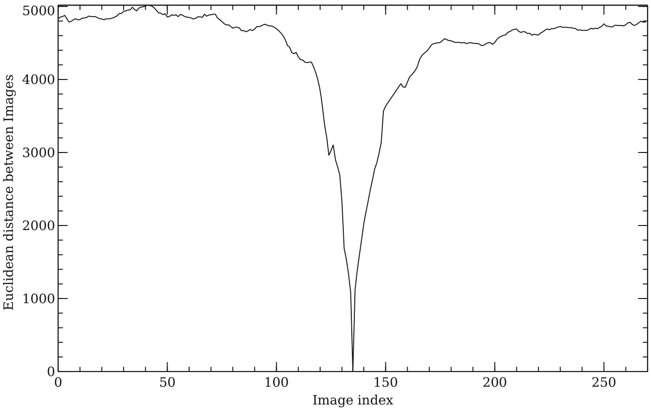


Fig. 2. Dissimilarity between center image and others images captured while the robot translated away from the center/reference image

catchment area can be exploited to make the robot navigate back to its centre [7] using a gradient descent method. This property is used to select nodes from the stream of images.

Nodes Creation/Spatial Sampling: The size of the catchment area is captured using the gradient of the Euclidean distance, a gradient of 0 indicating its maximum size. To ensure that nodes remain within the actual catchment area, a threshold m on the Euclidean distance gradient is used to select the next node from the previous node. A threshold of 0 retains the fewest nodes while a higher threshold increases the number of nodes in the map, controlling the map density. As can be seen in Fig. 2, there is noise in the Euclidean distance. This is due to noisy images but also the attitude of the robot changing on uneven terrain. A running average of the last four gradients is therefore used to filter the noise, as a compromise over efficiency and accuracy of the catchment area detection.

As mentioned above, the images are aligned using the Visual Compass [6]. It can happen that this method produces a suddenly drifting alignment. This is corrected by performing a local alignment (from the one calculated by the Visual Compass) between the previous node and the current image. Finally, to reduce the importance of the luminance information in the *Lab* colour space used by the Visual Compass over colour information, the L component was re-scaled so that its variance matches that of the colour information (components a and b).

The process is as follows. The first image is kept as a node. Subsequent images are compared to the previous node using the Euclidean distance and calculating the gradient of the distance. As the gradient falls below the threshold m, the corresponding image is used to create a new node. The process continues from that node and is repeated until all the images are processed. Figure 3 shows the Euclidean distance between images and the previous node, visible as a 0 distance.

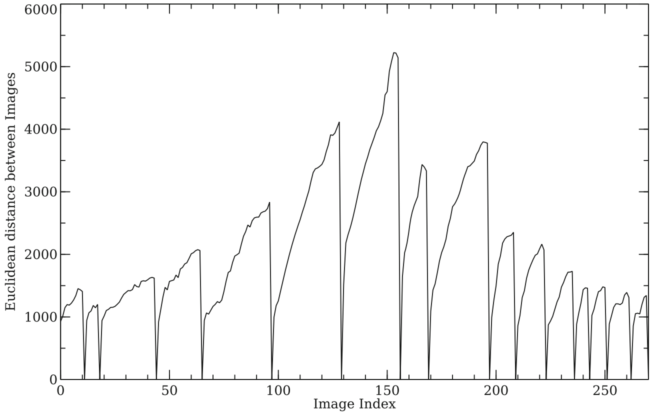


Fig. 3. Creation of nodes as the Euclidean distance to the previous node

In the sequence used for Fig. 3, the first part of the dataset corresponds to a grassy area, the middle part a road crossed by the robot and the final part another grassy area, shown by the differences in Euclidean distances.

Loop Closures: While adding nodes to the map, loop closures are sought for. To this effect, localisation is first performed (is the new node/location already in the map?), potentially followed by the creation of new links. The process involves comparing each newly added node to all existing nodes of that map and link them if the Euclidean distance between them is *low enough*. This is done using a threshold determined for each node based on local appearance. When a node j is created, its Euclidean distance to the previous node i is a measure of the catchment area for nodes i and j. A conservative threshold τ_j for loop closure detection with node j is therefore the minimum of all distances between node j and its neighbours in the stream of nodes:

$$\tau_j = \min\left(d(I_{j-1}, I_j), d(I_j, I_{j+1})\right). \tag{2}$$

At the time node j is created, node $j+1$ does not yet exist. Therefore initially $\tau_j = d(I_{j-1}, I_j)$ and is then updated when node $j+1$ is created using Eq. (2).

A loop between nodes i and j is closed if the distance between the corresponding images is lower than both thresholds:

$$d(I_i, I_j) < \begin{cases} \tau_i \\ \tau_j \end{cases}. \tag{3}$$

When a loop is closed, the corresponding edge is added to the map.

Loop closures are detected and created during the construction of the map. This implies that Eq. (3) can only be met in a two step process, before and after node $j+1$ is created, as described above. When node j is created, a list

Algorithm 1. Algorithm for Loop Closure Detection

```
procedure LoopClosureDetection($I_j, I_{j-1}, H_{j-1}$)
    ▷ $I_j$: new node, $I_{j-1}$: previous node, $H_{j-1}$: list of hypotheses about node $j-1$
    ▷ Check hypotheses for node $j-1$
    while $H_{j-1}$ no empty do
        $I_i$ = first image of $H_{j-1}$
        if $d(I_{j-1}, I_i) < \min(\tau_i, \tau_{j-1})$ then        ▷ Eq. 3 with updated $\tau_{j-1}$
            LoopClose($I_i$, $I_{j-1}$)
        end if
        remove $I_i$ from $H_{j-1}$
    end while
    ▷ Create hypotheses for node $j$
    $i = 0$                                                        ▷ Starting from first node
    $H_j = \emptyset$                                              ▷ With an empty list of hypotheses
    while $(i < j-1)$ do                                           ▷ Checking all nodes up to the previous one
        if $d(I_i, I_j) < \min(\tau_i, \tau_j)$ then              ▷ Eq. 3 with incomplete $\tau_j$
            $H_j = \{H_j, I_i\}$                                   ▷ Add $I_i$ to the set of hypotheses
        end if
        $i = i + 1$
    end while
end procedure
```

of hypotheses is created using the partial information about node j. This list contains the images that satisfy the (as yet incomplete) test in Eq. (3):

$$H_j = \{I_i : I_i \in M, d(I_j, I_i) < \min(\tau_i, \tau_j)\}, \tag{4}$$

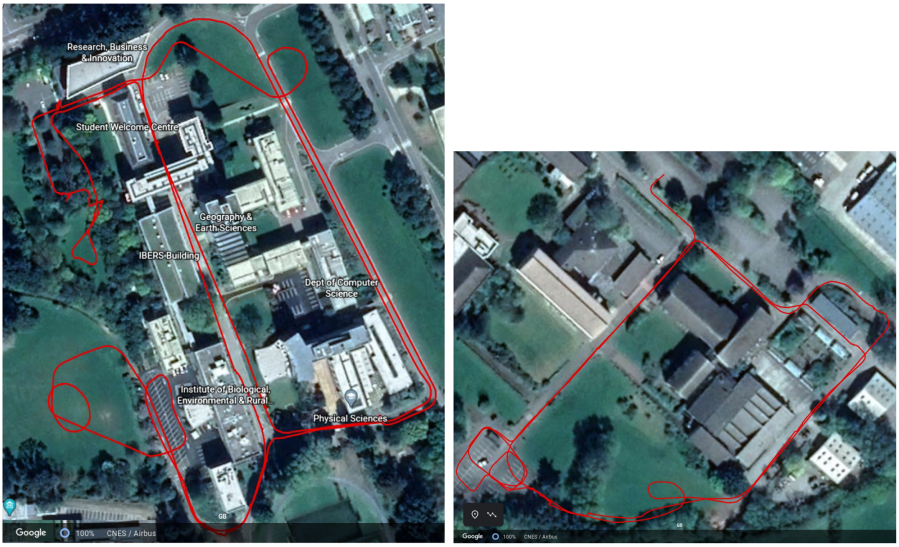
where M is the current map. When node $j+1$ is created, the threshold τ_j is updated and the hypotheses in the list H_j confirmed of not based on the new threshold. Algorithm 1 describes the process.

When a loop is closed, the corresponding loop closure thresholds are not updated to take into account the new edges added to the map. Indeed, doing so would result in reducing further and further the threshold (using the Eq. (2)), eventually stopping any new loop closures from being created.

The loop closure threshold in Eq. (2) corresponds to the smallest catchment the node. In some cases this could prevent a navigation strategy from reaching the node should the robot start from the edge of the catchment area. We therefore multiply the threshold τ with a constant $\gamma \leq 1$ that allows the specification of a smaller catchment area for loop closure. This also allows control of the trade-off between high false positives rates and low false negative rates.

3.3 Creating Nodes in Sequence Using Other Strategies

In Sect. 3 we described a method to adaptively select nodes from a stream of images. Other methods are possible, which we describe here and against which we compare our method.



(a) AberCampus00 dataset

(b) LlanCampus00 dataset

Fig. 4. Aerial view of the datasets

The simplest method is to use regular spatial sampling s. This is what is used in some state of the art work (e.g. [2]). The drawback with such method is that some areas could end up being over-sampled (such as long traverses of homogeneous terrain) or some important events might be missed (such as sudden transitions between two areas).

The second method is to use the reference images created by the Visual Compass [6] as the nodes of the map. The method automatically selects some of the images from the stream of images as reference images against which changes in orientation are calculated. This selection is based upon the quality of match between the reference image and successive images. That quality is the normalised amplitude of the Euclidean distance function expressed as the difference between best and worst match between two images. When that quality falls below a threshold β, the reference image is changed to the current image. See [6] for more detail. In this sampling method, high values of the threshold create a densely populated map while low values create fewer nodes.

4 Experimental Results

Multiple datasets were captured ranging over different lengths using one of our outdoors platforms (the Idris robot, a four wheel drive robot equipped with various sensors). The panoramic images have a resolution of 720 × 138. In other words, the angular resolution is 2 pixels per degree of rotation. We present here results on two datasets captured over paths 1000 m for LlanCampus00 2400 m for AberCampus00. These are shown in Fig. 4.

While mapping an environment, each node is stamped with its GPS coordinates. The GPS coordinates are only used to estimate the size of the catchment area in Cartesian space in order to evaluate the method as well as to visualise maps. This is computed using the physical distance traveled up to the point where the filtered gradient of Euclidean distance reaches zero (Fig. 2). Each node therefore has its own catchment area and any other node falling within it is considered as a TP (True Positive) loop closure. The loop closures with nodes outside the catchment area are considered FP (False Positives). Similarly, TN (True Negatives) are those tested connections that are not retained as loop closures and are outside of the catchment area the node. Finally, FN (False Negatives) are connections that are inside the catchment area but not detected as loop closures by the method.

The performance of loop closure detection is evaluated by the precision (P_r) and recall (R_e) of loop closures:

$$P_r = \frac{TP}{TP + FP} \tag{5}$$

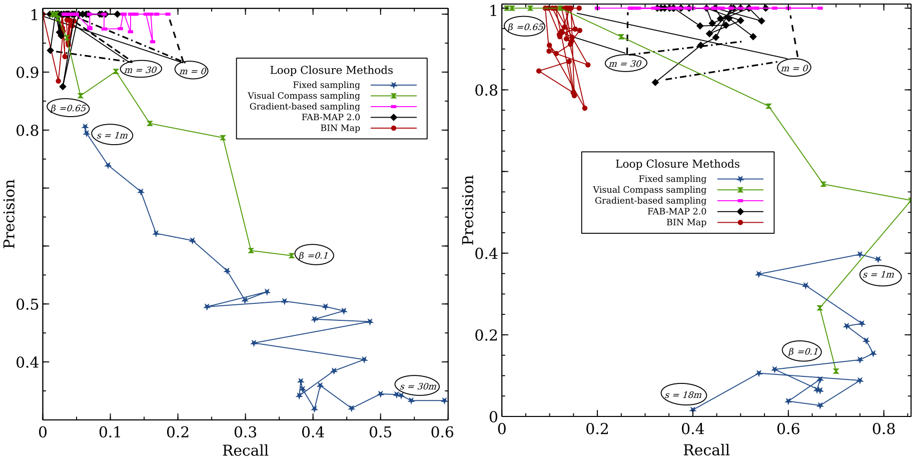
$$R_e = \frac{TP}{TP + FN} \tag{6}$$

Precision gives the proportion of the detected connections that are correct. The recall gives the proportion of correctly identified loop closures out of the total number of actual loop closures. A precision of 1 indicates that there are no falsely detected connections. A recall of 1 indicates that no actual connections are missed. A value of 0 for precision and recall indicates that no connections were detected. The trend in the literature is to increase recall while maintaining a precision of 1 [2,4].

Our method is compared with two well-known methods [2,4], which are discussed in Sect. 2. This comparison is based on the correctness of the loop closure detection. Since neither of these methods provides an automatic way of sampling the stream of images, we use our sampling method (gradient-based) to provide images to these two methods. Figure 5 is the plot of precision-recall for loop closures the datasets. The topological maps created using the gradient-based sampling method are shown in Fig. 6.

For the gradient-based sampling method we used values of the threshold m from 0 to 30 with increment of 1. For all results presented here, the multiplier γ was kept at 0.8. The threshold β for the Visual Compass sampling method was set from 0.1 to 0.65 with increments of 0.05. The fixed spatial sampling s was set 1 m 30 m with increments of 1m, but limited 18 m for LlanCampus00 because no loop closures were detected beyond that sampling, the dataset being smaller. For both FAB-MAP 2.0 and BIN Map, loop closures were selected if their probability was higher than 0.99.

The results in Fig. 5 show that precision and recall of the gradient-based sampling method performs better for all threshold values compared to the other methods (maximum precision, high recall). That is why gradient-based sampled data (same range of the m) was used with the other methods [2,4] for comparison.



(a) AberCampus00 dataset

(b) LlanCampus00 dataset

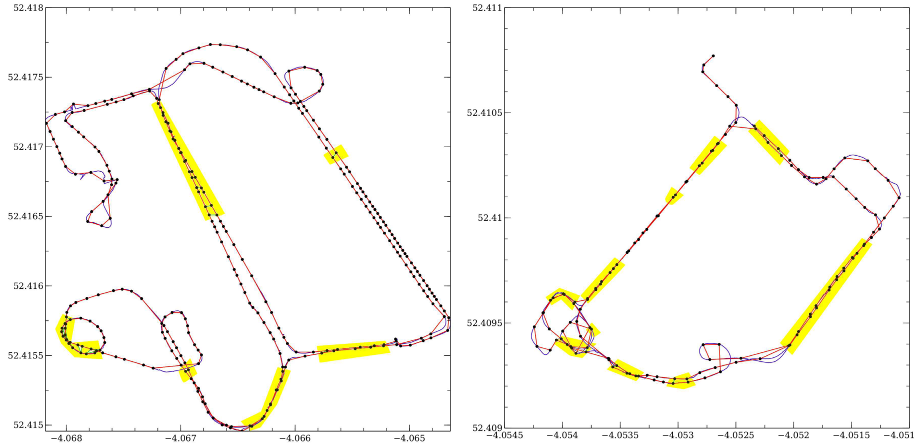
Fig. 5. Loop closure precision and recall for various methods. The extreme values of the control parameters (m, β and s) are indicated in ovals with lines pointing to the corresponding ends of the plots, with dashes for the gradient-based sampling method, dot-dash for FAB-MAP 2.0 and solid for BIN Map.

For the AberCampus00 dataset, both methods in [2,4] have high precision but low recall compared to the gradient-based sampling method, which has high recall when the map is sparsely (lower m value) populated and low recall when densely (higher m value) populated. In densely populated maps, the Cartesian space distance between nodes is shorter, resulting in a lower threshold τ (the Euclidean distance at the point of creation of the node. This in turn reduces the effective catchment area of nodes for loop closure detection. Coupled with the increased number of actual loop closures (denser nodes imply more loop closures), the detection rate (TP) decreases, resulting in a lower recall.

The results for the LlanCampus00 dataset show a higher recall than for AberCampus00, for all methods. This is due to most of the parallel paths being detected as loop closures for the former. In particular a long stretch of parallel paths in the AberCampus00 dataset has not been detected as loop closures. This is due to large image differences between the two paths because of close proximity to grass on one side and bushed/buildings on the other, both sides being different in colour. This is visible in Figs. 4 and 6.

The results using the FAB-MAP 2.0 and BIN Map methods are not consistent when varying m; this is likely due to the fact that the sampling used was not based on the features used by the methods. FAB-MAP 2.0 performs better than BIN Map on both datasets.

The loop closure (localisation) time complexity is linear in the number of nodes in the map since a comparison is made with all existing nodes for each



(a) Topological map for AberCampus00 (b) Topological Map for LlanCampus00

Fig. 6. Topological maps for AberCampus00 and LlanCampus00. The green connections (highlighted in yellow) are loop closures while the red connections correspond to the creation sequence. These were produced with $m = 10$.(Color figure online)

newly added node. For a map of 175 nodes, the loop closure computation time on an Intel core $i7$ (6^{th} generation) is 400 ms.

Both fixed and Visual Compass sampling methods can produce higher recalls at the expanse of lower precision. This is due to these methods producing sparser maps at settings of lower sampling rates of their parameters ($s = 30$ and $\beta = 0.1$). This behaviour is similar to that of the gradient-based sampling method at low values of the gradient threshold (large distance between nodes). These two methods however never reach a precision of 1. This may be due to a spatial sampling not adequate at the other end of the parameter range. In any case, the sampling not being adapted to the information used in the loop closures, it is unlikely that such method will perform well. Note that the behaviour of the fixed sampling method on the LlanCampus00 dataset is erratic, probably due to the relative small size of the environment/dataset and that fixed sampling can randomly select nodes that are nearby or not at key locations.

As can be seen on the top right corner of the map in Fig. 6b, some loop closures at junctions of paths are missed. This is due to nodes not being created in synchronisation between the multiple branches of the junction.

5 Conclusion and Future Work

In this paper, we have presented a novel appearance-based visual topological mapping method, which uses an adaptive sampling method to select relevant images and creates loop closures that are based on local properties of the map (changes of the local appearance). The method uses panoramic images captured at regular intervals while traversing the environment of the robot. The method was evaluated here using precision and recall. Other metric need to be devised to measure the quality of the produced topologies.

The purely gradient-based sampling method adapts to rapidity of change of the environment but sometimes creates nodes that prevent some loop closures from being created.

One major drawback of our approach is the time complexity associated to loop closures and localisation. Work is in progress to build hierarchies onto the map to reduce the time complexity allowing larger environments to be covered.

References

1. Cummins, M., Newman, P.: FAB-MAP: Probabilistic localization and mapping in the space of appearance. Int. J. Robot. Res. **27**(6), 647–665 (2008)
2. Cummins, M., Newman, P.: Appearance-only SLAM at large scale with FAB-MAP 20. Int. J. Robot. Res. **30**(9), 1100–1123 (2011)
3. Garcia-Fidalgo, E., Ortiz, A.: iBoW-LCD: an appearance-based loop-closure detection approach using incremental bags of binary words. IEEE Robot. Autom. Lett. **3**(4), 3051–3057 (2018)
4. Garcia-Fidalgo, E., Ortiz, A.: Vision-based topological mapping and localization methods: a survey. Robot. Auton. Syst. **64**, 1–20 (2015)
5. Ismail, K., Liu, R., Zheng, J., Yuen, C., Guan, Y.L., Tan, U.: Mobile robot localization based on low-cost LTE and odometry in GPS-denied outdoor environment. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2338–2343 (2019)
6. Labrosse, F.: The visual compass: performance and limitations of an appearance-based method. J. Field Robot. **23**(10), 913–941 (2006)
7. Labrosse, F.: Short and long-range visual navigation using warped panoramic images. Robot. Auton. Syst. **55**(9), 675–684 (2007)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
9. Lowry, S., et al.: Visual place recognition: a survey. IEEE Trans. Robot. **32**(1), 1–19 (2015)
10. Neal, M., Labrosse, F.: Rotation-invariant appearance based maps for robot navigation using an artificial immune network algorithm. In: Proceedings of the 2004 Congress on Evolutionary Computation, vol. 1, pp. 863–870 (2004)
11. Ranganathan, A., Menegatti, E., Dellaert, F.: Bayesian inference in the space of topological maps. IEEE Trans. Robot. **22**(1), 92–107 (2006)
12. Ray, A.K., Behera, L., Jamshidi, M.: GPS and sonar based area mapping and navigation by mobile robots. In: 2009 7th IEEE International Conference on Industrial Informatics, pp. 801–806 (2009)
13. Ulrich, I., Nourbakhsh, I.: Appearance-based place recognition for topological localization. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 1023–1029 (2000)
14. Werner, F., Maire, F., Sitte, J.: Topological SLAM using fast vision techniques. In: Kim, J.-H., et al. (eds.) FIRA 2009. LNCS, vol. 5744, pp. 187–196. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03983-6_23

15. Woodland, A., Labrosse, F.: On the separation of luminance from colour in images. In: Proceedings of the International Conference on Vision, Video, and Graphics, pp. 29–36. University of Edinburgh (2005)
16. Zhang, H.: BoRF: Loop-closure detection with scale invariant visual features. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3125–3130 (2011)