

Efficient Topological Mapping with Image Sequence Partitioning

Hemanth Korrapati, Youcef Mezouar, Philippe Martinet

LASMEA, CNRS JOINT UNIT 6602, 24, Avenue des Landais, 63177-AUBIÈRE, FRANCE

Email: firstname.lastname@lasmea.univ-bpclermont.fr

Abstract—Topological maps are vital for fast and accurate localization in large environments. Sparse topological maps can be constructed by partitioning a sequence of images acquired by a robot, according to their appearance. All images in a partition have similar appearance and are represented by a node in a topological map. In this paper, we present a topological mapping framework which makes use of image sequence partitioning (ISP) to produce sparse maps. The framework facilitates coarse loop closure at node level and a finer loop closure at image level. Hierarchical inverted files (HIF) are proposed which are naturally adaptable to our sparse topological mapping framework and enable efficient loop closure. Computational gain attained in loop closure with HIF over sparse topological maps is demonstrated. Experiments are performed on outdoor environments using an omnidirectional camera.

Index Terms—Topological Mapping, Omnidirectional Vision, Loop Closure

I. INTRODUCTION

Mapping is one of the fundamental problems of Autonomous Mobile robotics. Mapping problem can be widely categorized as Topological and Metrical [16]. Metrical mapping involves accurate position estimates of robots and landmarks of the environment. Topological mapping on the other hand represents an environment as a graph in which nodes correspond to places and the edges between them indicate some sort of connectivity. Recently, a third category called Topo-Metric Mapping [17], [9] is gaining popularity. Topo-Metric mapping is a hybrid approach which uses both metrical and topological information in map building. Building an accurate map either metrical or topological depends on loop closure accuracy. Such maps are difficult to build using metrical information which is prone to gross errors in position estimation of robot and landmarks. Topological maps facilitate accurate loop closure as they depend on appearance information rather than on exact metrical information of the environment.

Many powerful loop closing techniques for topological maps have been introduced recently [2, 3, 5, 7]. Most of them produce dense topological maps, in which every acquired image stands as a node in the topological graph. Sparser topological maps can be built by representing sets of contiguous images with similar appearance features as places. Each place is represented by a node in the topological graph. We refer to this kind of partitioning of image sequences into places as Image Sequence Partitioning (ISP). In a sparse topological map, since each node represents multiple images, fewer nodes would be sufficient for the map representation. Maps with

fewer nodes reduce computational complexity involved in loop closure and map merging. An example map merging problem can be to localize a topological map of a tiny environment in a larger map (ex:- google maps).

We use a topological mapping framework which facilitates coarse loop closure to the node level and a finer loop closure to the image level. The topological map is represented as a graph $T = (N, E)$, where N and E are sets of nodes and edges respectively. The map is updated with each newly acquired image. Every new image (query image) is verified if it is similar to a previously visited node (place) or the current place node and if so, the corresponding node is augmented with the image. If the query image is not similar to any of the existing nodes, then a new place node is created and augmented with the image. This process of map update is nothing but Image Sequence Partitioning (ISP). Each node contains a set of representative features representing all the member images of the node (place). The representative features are used in evaluating node-image similarity during ISP.

Another contribution of this paper is the proposal of Hierarchical inverted files (HIF) for efficient loop closure at both node and image levels. Similar to traditional inverted files used for object recognition [11] and loop closure problems [2], HIFs are also associated to the visual words in the vocabulary. HIFs combine the power of regular inverted files with our sparse topological map structure and help in fast loop closure. Considering the fact that in a sparse topological map, images are again grouped into nodes, HIFs organize

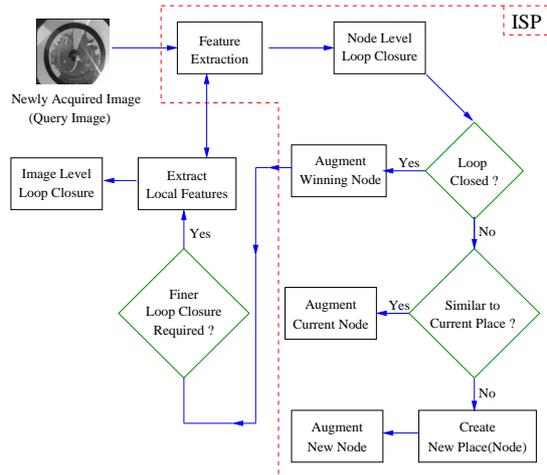


Fig. 1: A global view of our topological mapping framework.

the previous occurrence information of visual words in a hierarchical fashion which enables fast loop closure. We use an inverse similarity score evaluation methodology in order to take advantage of HIFs for loop closure.

Experiments were performed on omnidirectional image data acquired in outdoor urban environments. Omnidirectional images offer a 360 degree field of view which helps in building topological maps, invariant of robot's heading. As a result loop closure can be performed even if the robot is not headed in the same direction as of the previous visit to the same location. Map building in outdoor environments is challenging due to illumination variation and possible occlusions [15]. Sparsity and accuracy of topological maps constructed using ISP are evaluated. The computational savings achieved in HIF-based loop closure is analysed.

The rest of the paper is organized as follows: Section II details the related work done in this area. Section III describes the steps involved in ISP in detail and provides algorithmic illustrations. Section IV introduces HIFs and discusses how node and image level loop closures are performed using HIFs. Section V evaluates sparsity of maps produced by ISP, accuracy and computational cost of loop closure on the generated topological maps.

II. RELATED WORK

Scene Change Detection and Key Frame Selection for video segmentation and abstraction [19], [13] have similar goals as that of ISP. They try to represent a video with fewer images called key frames whenever there is a sufficient change in the scene and most of them focussed on video compression domain. The major difference between these video abstraction problems and topological mapping is that topological mapping demands localization of a query image which is obtained at a previously visited place, but with variation in illumination and viewpoint, and a possible occlusion. Hence, video segmentation techniques using pixel-wise intensity measures and global image features like histograms, motion based segmentation cannot be applied to our problem.

Loop closure in topological maps has gained popularity among mobile robotic researchers during the recent times. Many loop closure algorithms for topological mapping have been proposed and tested in both indoor [7], [2], [3], [21], [8] and outdoor environments [5], [6], [1], [12].

In [21], [22] topological maps are built for indoor environments. They segment the topological graph of the environment using normalized graph-cuts algorithm resulting in subgraphs corresponding to convex areas in the environment. In [8] SIFT features were used to perform matching over a sequence of images. They detected transitions between individual indoor locations depending on the number of SIFT features which can be successfully matched between the successive frames. In [14] fingerprint of an acquired image is generated using omnidirectional image and laser readings, and these fingerprints are compared to those of the previous images. If the similarity is above a threshold the image is added to the existing node and if not a new node is formed. All of these works were focused on indoor environments. Indoor environments contain convex

spaces (rooms) which are relatively simpler to be partitioned as compared to outdoor environments.

A topological mapping framework using incremental spectral clustering has been presented in [20]. Nodes containing similar images are constructed using incremental spectral clustering over the affinity matrix of the images, thereby producing a topological graph. An optical flow based ISP technique was presented in [12] for topological mapping in outdoor environments using a quad rotor robot. Optical flow is used to discover change in environmental appearance. In [1], gist features [18] were used to cluster images with similar appearance for topological map construction.

III. IMAGE SEQUENCE PARTITIONING

Figure 1 depicts a global view of our framework, in which we can see a modular view of ISP enclosed by a red dashed line. As can be seen from Figure 1, ISP consists of three main modules: node level loop closure, evaluation of similarity to current place and new node formation. Given a query image, initially SURF [4] features are extracted from the image. Using the SURF features, we evaluate the node-image similarity of the query image with all the nodes in the graph except the current place node and pick out the top k similar nodes. The top k similar nodes are assigned to the set of winning nodes N_w . This process is called node level loop closure as it finds the previously visited places (nodes) most similar to the query image. Only the representative feature sets of the nodes are used to compute the node-image similarities during node level loop closure. In our framework, the representative features of a node are the SURF features of the first image augmented to the node.

An empty N_w indicates loop closure failure ; that is, query image is not similar to any of the previously visited places. In that case, query image similarity to the current node is evaluated. If the similarity of query image with current place node is above a certain threshold, current node is augmented with the query image. If the query image is not similar to current node also, a new node is created with the query image. But if N_w is not empty indicating a loop closure, then the set of winning nodes can be considered for a thorough image level loop closure.

Algorithm 1 shows the steps involved in ISP. The 'node_level_loop_closure' function in lines 4 is discussed in detail in sections IV. The function 'current_node_image_similarity' evaluates the similarity of the current node with that of the query image. This is done by matching the SURF features of the query image to that of the features of the current place node. Feature matching is performed as proposed in [10]. Two features are considered to be matched if the ratio of the best match and the second best match is greater than 0.6.

IV. LOOP CLOSURE & HIERARCHICAL INVERTED FILES

Node and Image level loop closures are performed at image level using visual words corresponding to the SURF features of the image. Given a query image, to find the most similar

Algorithm 1 Image Sequence Partitioning Algorithm

```

1: procedure PROCESS_QUERY_IMAGE( $\mathbf{T}, I_q, n_c$ )  $\triangleright \mathbf{T}, I_q, n_c, \mathbf{T.N}$  are the
   Topological graph, query image, current node in topological graph & node-set of
   topological graph respectively.
2:    $\mathbf{N}' = \mathbf{T.N} - \{n_c\}$   $\triangleright$  Reference node set excluding current node.
3:    $\mathbf{N}_w = \{\}$   $\triangleright$  Set of winning nodes.
4:    $\mathbf{N}_w = \text{Node\_Level\_Loop\_Closure}(\mathbf{N}', I_q, Th_s)$ 
5:   if is_empty( $\mathbf{N}_w$ ) then
6:     if current_node_image_similarity( $n_c, I_q$ ) >  $Th_t$  then
7:        $n_c.add\_image(I_q)$ 
8:     else
9:        $n = \text{new\_node}()$ 
10:       $n.add\_image(I_q)$ 
11:      update_map( $\mathbf{T}, n$ )
12:    end if
13:   else
14:      $\mathbf{N}_w = \text{get\_top\_k\_similar\_nodes}(\mathbf{N}_w)$ 
15:      $\mathbf{I}_w = \text{get\_images\_of\_nodes}(\mathbf{N}_w)$ 
16:      $n^* = \text{Image\_Level\_Loop\_Closure}(\mathbf{I}_w, \mathbf{N}_w, I_q)$ 
17:     add_image_to_node( $I_q, n^*$ )
18:     update_map( $\mathbf{T}, n^*$ )
19:   end if
20: end procedure

```

reference image in the database, [11, 7] uses a Tf-Idf based similarity score for all the reference images in the database and the query image and select the reference images corresponding to the top n similarity scores. We use an inverse methodology to compute image similarities for loop closure. The steps involved are enumerated as follows:

- 1) Let the set of reference images be $\mathbf{I} = \{I_1, I_2, \dots, I_M\}$. We consider a histogram H with the number of bins corresponding to the number of reference images, M .
- 2) Extract the set of visual words $\mathbf{W} = \{w_1, w_2, \dots, w_p\}$ from the query image, I_q .
- 3) For each visual word w_i , using the inverted file IF_i of the word, we extract the reference image indexes $\mathbf{I}^{w_i} = \{I_1^{w_i}, I_2^{w_i}, \dots\}$ in which the word has been previously seen. The histogram bins corresponding to these extracted reference images are incremented by a factor of Tf-Idf of the corresponding word.

$$H[I_j^{w_i}] = H[I_j^{w_i}] + Tf - Idf(I_j^{w_i}, w_i) \quad (1)$$

The resulting histogram can be interpreted to contain the degrees of similarity of the query image with respect to the reference images. As we can see, the loop closure computation time only depends on number of words in the query image and the average inverted file length at that instant. As a result loop closure time does not increase so steeply as is the case with forward method. A closely related work can be found in [6, 2, 3]. But this method is suitable only for loop closure over dense topological maps but not for sparse topological maps in which each node represents multiple images. With a change in the inverted file structure, we can adapt this similarity evaluation method to sparse topological maps.

A regular inverted file corresponding to a visual word simply contains a list of all previous images which contained the word. We associate Hierarchical inverted files (HIF) to each visual word. As the name suggests, HIF contain two levels. The first level consists of the ids of the nodes in which the visual word occurred previously. The second level consists of small child inverted files attached to each of the

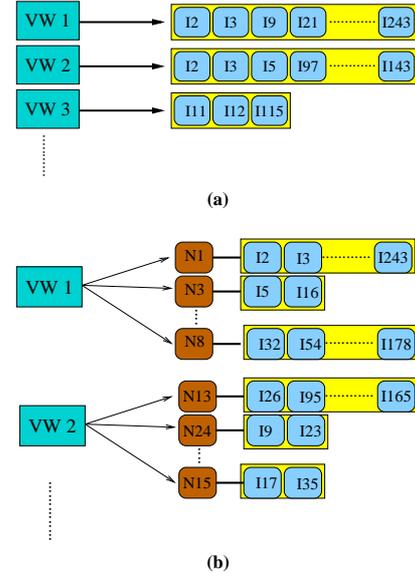


Fig. 2: (a) represents a traditional inverted file. (b) represents a hierarchical inverted file. VW1, VW2,... represent visual words. N1, N2,... represent node ids in the first level of HIF and I1, I2, I3, ... represent the image ids in the inverted file.

node id. These image ids indicate the images belonging to the parent node in which the visual word has occurred. Each child inverted file corresponding attached to a node of the topological graph, contains the list of all previous images belonging to the parent node in which the word has occurred. The difference between traditional inverted files and HIFs is illustrated in figure 2. To perform a node level loop closure using HIF, we do not have to go through the entire HIF, but its sufficient to go through first level (node ids) of the HIFs. For an image level loop closure using HIF, we only have to traverse through those child inverted files corresponding to the winning nodes; which form only a fraction of the total HIF. Thus, HIFs offer computational gain in loop closure when compared to regular inverted files which is demonstrated in section V. Algorithms 2 and 3 give a clearer picture of the node level and image level loop closures. There can be multiple winning images given by the image level loop closure and hence multiple corresponding nodes, but for the sake of simplicity we do not represent that in the algorithm. In such cases we use RANSAC based geometric verification to find the right match.

Algorithm 2 Node Level Loop Closure Algorithm

```

1: procedure NODE_LEVEL_LOOP_CLOSURE( $\mathbf{N}', I_q, Th_s$ )
2:    $\mathbf{N}' = \mathbf{T.N} - \{n_c\}$ 
3:    $H = \text{Histogram}(\text{size\_of}(\mathbf{N}'))$ 
4:    $\mathbf{W} = \text{extract\_and\_quantize\_Features}(I_q)$ 
5:   for each word  $w_i$  in  $\mathbf{W}$  do
6:      $HIF_i = \text{get\_hierarchical\_inverted\_file}(w_i)$ 
7:     for each node  $n_j$  in  $HIF_i$  do
8:        $H[n_j] = H[n_j] + 1$ 
9:     end for
10:  end for
11:   $\mathbf{N}^* = \text{get\_winners\_from\_histogram}(H, Th_s)$ 
12:  return  $\mathbf{N}^*$ 
13: end procedure

```

Algorithm 3 Image Level Loop Closure Algorithm

```

1: procedure IMAGE_LEVEL_LOOP_CLOSURE( $I_w, N_w, I_q$ )
2:    $H = \text{Histogram}(M)$ 
3:    $\mathbf{W} = \text{extract\_and\_quantize\_Features}(I_q)$ 
4:   for each word  $w_i$  in  $\mathbf{W}$  do
5:      $HIF_i = \text{get\_hierarchical\_inverted\_file}(w_i)$ 
6:     for each node  $n_j$  in  $N_w$  do
7:        $IF_i^{n_j} = \text{get\_inverted\_file\_of\_node}(HIF_i, n_j)$ 
8:       for each entry  $f_i$  in  $IF_i^{n_j}$  do
9:          $H[f_i] = H[f_i] + Tf - Idf(f_i, w_i)$ 
10:      end for
11:    end for
12:  end for
13:   $I^* = \text{get\_winner\_from\_histogram}(H)$ 
14:   $n^* = \text{get\_corresponding\_node}(I^*)$ 
15:  return  $n^*$ 
16: end procedure

```

V. EXPERIMENTS

Our experimental setup consists of a Pioneer P3DX robot equipped with an omnidirectional camera. A laptop equipped with an Intel Centrino 2 processor running ubuntu 9.04 is used for data processing. The experiments were carried out in our artificial urban environment - PAVIN. The environment contains roads, artificial buildings, and a variety of real-world road settings like junctions, traffic lights, round-abouts, curved roads and dead ends.

Omnidirectional images were acquired at a frame rate of 2 fps, as the robot moves along a manually controlled trajectory. Image data was acquired in four installments (A, B, C and D) at very different times of two days and hence contained significant illumination variation. Figure 3 shows the parts of the environment through which the robot traversed during each installment. We took care that data from all the four installments contained overlaps so as to put our loop closure algorithm to test.

We constructed two data-sets by combining data from all the four installments. Dataset-6560 was obtained by combining data of installments A, C and D. It contains 6560 images with 52 possible loop closures. Another data-set Dataset-11200 was obtained by a combination of all the four installments. It contains 11200 images and 71 possible loop closures. The number of loop closures were determined by manually examining the data-sets.

A. ISP - Sparsity

The number of nodes in a topological map indicate its sparsity. An ideal topological map is one in which each distinct place in the environment is represented by a node in the topological graph. The sparsity of these ideal maps represents the optimal sparsity of the actual environment. But practically, an ideal topological map is far from being attained. Different features produce different topological representations of the environment. We have experimented using SURF128, U-SURF128, and SIFT features, out of which we found out that U-SURF128 features lead to topological structure closest to the ground truth.

Another important factor that effects stability of appearance is the image distortion. The features directly extracted from warped omnidirectional images are unstable as the appearance of keypoint patch changes very much even with a small

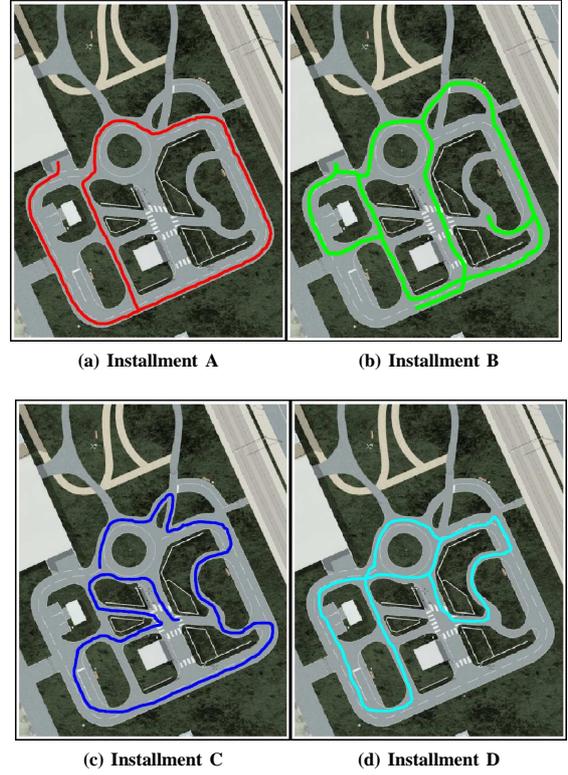


Fig. 3: Shows paths traversed by the robot during each image acquisition installment.

TABLE I: SPARSITY

(a) Sparsity - Warped			
	SURF128	U-SURF128	SIFT
DATASET-6560	539	502	1582
DATASET-11200	756	742	2795

(b) Sparsity - Unwarped			
	SURF128	U-SURF128	SIFT
DATASET-6560	504	473	1037
DATASET-11200	737	723	1257

displacement of the camera. Hence it is likely that the maps produced using warped images contain greater number of place nodes. This happens because due to the feature instability, each place can be understood as multiple adjacent places and hence multiple nodes in the topological graph. The undistorted (unwarped) images produce relatively sparser maps. Tables I(a) and I(b) show the sparsity of maps produced by ISP using different features on warped and unwarped images. We can see that U-SURF128 is the best performing feature producing the most sparse maps.

B. Accuracy

In this subsection, we discuss the accuracy of the maps produced by ISP, based on the number of accurate loop closures and the obtained false positives. The most sparse map may not guarantee an accurate map. Only those maps with accurate place partitioning are accurate and can lead to accurate loop closures. Thus a good mapping technique is

TABLE II: NODE LEVEL LOOP CLOSURE ACCURACY

	#(LC)	#(FP)
Dataset-6560	49	4
Dataset-11200	68	6

one which provides an optimal combination of sparsity and accuracy.

Given a query image, first we perform loop closure at node level and then at image level if more accuracy is required. Table II shows the number of loop closures detected and the number of false positives obtained on Dataset-6560 and Dataset-11200 respectively by node level loop closure. We can see that a few loop closures are missed out, and some false positives are observed. The inaccuracy can be attributed to the imperfections in ISP. There is a direct relation between accuracy of the maps and ISP. The way in which we perform topological mapping does not induce any information loss. In other words we do not perform any kind of sampling or selection of reduced number of features. Instead, we consider each and every feature extracted from the images and store it in HIFs. This process guarantees that there is no information lost during the mapping process. But inaccurate loop closure detection might occur due to inaccurate partitioning of places. For example a place can be represented by two nodes by partitioning it inaccurately during ISP due to appearance feature instability. As a result, during node level loop closure using a query image, both of the nodes representing that place may not get high similarity scores and hence the loop closure becomes inaccurate. A good ISP algorithm produces maps with minimum number of these situations.

Image level loop closure accuracy depends on the accuracy of node level loop closure. If node level loop closure selects an inaccurate set of winning nodes, then as a result, image level loop closure also becomes inaccurate. However in case of an accurate node level loop closure, we have observed that 99% accuracy was possible in image level loop closure irrespective of the ISP technique. Figure 4 shows two loop closure scenarios that occurred in our mapping.

C. Computational Time

Table III shows average computational time (in milliseconds) taken by each stage of our topological mapping framework in processing each query frame on both Dataset-6560 and Dataset-11200. The abbreviations NLLC, LFE+QUANT, ILLC stand for Node Level Loop Closure, Local Feature Extraction & Quantization and Image Level Loop Closure respectively. NLLC is the node level loop closure which involves extracting the most similar nodes using HIFs as mentioned in Algorithm 2. This takes 10ms as shown in table III, and requires additional 50ms in order to compare with the current place node whenever needed. Obviously, local feature extraction (LFE) (200ms) and quantization (QUANT) (70ms) time is constant for every acquired frame. Actually, time required for both of these tasks increases with the number of features in an image.

Computation time of image level loop closure (ILLC) is too low. This low computation time is the result of using

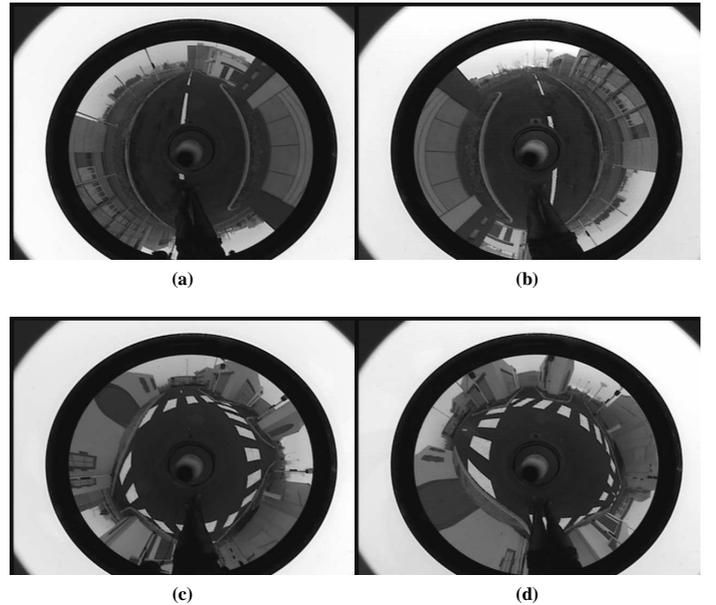


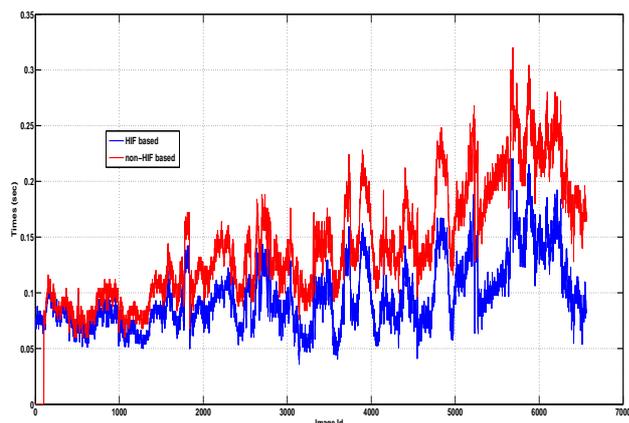
Fig. 4: Example loop closures.

TABLE III: AVERAGE COMPUTATION TIMES (in ms)

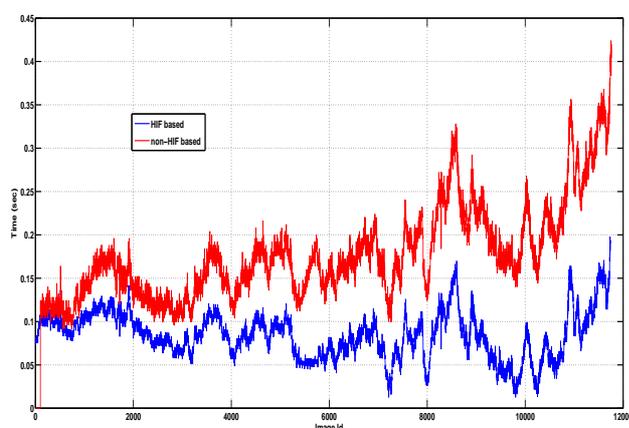
NLLC	LFE+QUANT	ILLC
10 + 50	200 + 70	21

hierarchical inverted files(HIF). As we mentioned before, HIFs make the loop closure computation almost independent of the number of reference images and also in our case, nodes of the topological graph.

Figures 5(a) and 5(b) show graphs comparing the loop closure times of our HIF-based method and without using HIF (non-HIF based). Red curves in the graphs indicate the time taken for feature quantization, node level loop closure and image level loop closure, for each image frame in a sequence. The blue curves represent the time taken by feature quantization and similarity score generation using inverted files by using inverse similarity evaluation methodology. We can see that the loop closure time of non-HIF based method increases relatively more with the increase in the number of images in the map, while our method using our method, loop closure time increases more slowly. Also, the performance gain becomes more prominent in case of huge datasets (huge number of images) as can be seen in the figure 5(b) corresponding to Dataset-11200, which contains 11200 images. The non-HIF based loop closure time for Dataset-11200 increases less steeply than that of Dataset-6560. This happened because the average number of features of Dataset-11200 is lesser than that of Dataset-6560 and as a result it takes lesser time to process each reference frame. This efficiency of our HIF-based method can be attributed to the combination of sparse topological mapping and HIFs for efficient map storage. The representational power of HIFs saved lot of computation involved in loop closure.



(a) Loop Closure time - Dataset-6560



(b) Loop Closure time - Dataset-11200

Fig. 5: Loop closure computation times of non-HIF based loop closure and HIF based loop closure on maps generated on our datasets.

VI. CONCLUSION

We proposed a sparse topological mapping framework involving two levels of loop closure. Hierarchical Inverted Files(HIF) were naturally adaptable for loop closure in our sparse topological mapping framework and made fast loop closure possible. Image Sequence Partitioning(ISP) played a key role in producing sparse topological maps. Sparsity of the maps produced by different features are analyzed and the accuracy is evaluated. Finally, our framework is evaluated on computational time required for loop closure. The experiments prove our argument that HIFs are suitable for sparse topological maps as they take advantage of the sparsity of the map in performing loop closure efficiently without discarding any information.

ACKNOWLEDGMENT

This work has been supported by the ANR projects - CityVip and R-Discover. Special thanks to Mr.Jonathan Courbon for his help in experimentation.

REFERENCES

- [1] J. Kosecka A. C. Murillo, P. Campos and J. J. Guerrero. Gist vocabularies in omnidirectional images for appearance based mapping and localization. In *10th IEEE Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS), held with Robotics, Science and Systems*, 2010.
- [2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008.
- [3] Adrien Angeli, Stéphane Doncieux, Jean-Arcady Meyer, and David Filliat. Visual topological slam and global localization. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation, ICRA'09*, pages 2029–2034, 2009.
- [4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110, June 2008.
- [5] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- [6] Mark Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics Science and Systems (RSS)*, Seattle, USA, June 2009.
- [7] Friedrich Fraundorfer, Christopher Engels, and David Nistér. Topological mapping, localization and navigation using image collections. In *IROS*, pages 3872–3877, 2007.
- [8] Jana Koseck, Fayin Li, and Xialong Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27 – 38, 2005.
- [9] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid Spatial Semantic Hierarchy. In *Proceedings of the IEEE Intl. Conf. on Robotics & Automation (ICRA-04)*, 2004.
- [10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [11] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06*, pages 2161–2168, 2006.
- [12] Navid Nourani-Vatani and Cedric Pradalier. Scene change detection for topological localization and mapping. In *IEEE/RSJ Intl. Conf. on Intelligent Robotics and Systems (IROS)*, 2010.
- [13] Cees G.M. Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*, 25:5–35, 2005.
- [14] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. In *IEEE/RSJ Intl. Conf. on Intelligent Robotics and Systems (IROS)*, pages 2429–2434, 2005.
- [15] Sebastian Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [17] Nicola Tomatis. Hybrid, metric-topological representation for localization and mapping. In *Robotics And Cognitive Approaches To Spatial Mapping*, volume 38 of *Springer Tracts In Advanced Robotics*, pages 43–63, 2008.
- [18] Antonio Torralba, Kevin P. Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV*, pages 273–, 2003.
- [19] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3, 2007.
- [20] Christoffer Valgren, Tom Duckett, and Achim Lilienthal. Incremental spectral clustering and its application to topological mapping. In *In Proc. IEEE Int. Conf. on Robotics and Automation*, pages 4283–4288, 2007.
- [21] Zoran Zivkovic, Olaf Booij, and Ben Kröse. From images to rooms. *Robot. Auton. Syst.*, 55:411–418, May 2007.
- [22] Z.Zivkovic, B.Bakker, and B.Krose. Hierarchical map building using visual landmarks and geometric constraints. In *In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7–12, 2005.