SAIL-MAP: Loop-closure detection using saliency-based features

Conference Paper · September 2014

DOI: 10.1109/IROS.2014.6943206

4 authors:

Merwan Birem
Flanders Make
5 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Jean-Charles Quinton
Université Grenoble Alpes
38 PUBLICATIONS **73** CITATIONS
SEE PROFILE

François Berry
Université Clermont Auvergne
88 PUBLICATIONS **371** CITATIONS
SEE PROFILE

Y. Mezouar
SIGMA-Clermont
152 PUBLICATIONS **1,509** CITATIONS
SEE PROFILE

SAIL-MAP : Loop-Closure Detection Using Saliency-Based Features

Merwan BIREM[1], Jean-Charles QUINTON[1], François BERRY[1], and Youcef MEZOUAR[1]

Abstract— **Loop-closure detection, which is the ability to recognize a previously visited place, is of primary importance for robotic localization and navigation problems. We here introduce SAIL-MAP, a method for loop-closure detection based on vision only, applied to topological simultaneous localization and mapping (SLAM). Our method allows the matching of camera images using a novel saliency-based feature detector and descriptor. These features have been designed to benefit from the robustness to viewpoint change and image perturbations of bio-inspired saliency algorithms. Additionally, the same algorithm is used for the detector and descriptor. The results obtained on different large-scale data sets demonstrate the efficiency of the proposed solution for localization problems.**

I. INTRODUCTION

Visual mapping is a fundamental problem of mobile robotics, since other sources of information might sometimes be unreliable (e.g. GPS). Due to the vast scale of current real world mapping problems, topological approaches are gaining ground against metric ones. Topological maps indeed facilitate the representation and management of sensor data due to their explicit independence to metric information.

Loop-closure detection is the problem of knowing if the robot is revisiting an already visited area of the environment, and thus plays a central role in accurate map construction. Many powerful approaches have been proposed recently to address this problem efficiently [1], [2], [3], [4]. When a robot is navigating in its environment, it may roughly pass through the same place several times, but will actually never observe the environment under the exact same conditions (brightness, pose, mobile objects...). This is why a good loop-closure detector (LCD) must take into consideration the inherent variability in such parameters.

The first step toward the development of a LCD is to choose between on-line and off-line methods. Both are heavily discussed in the literature, since the choice has significant consequences, and generally depends on the goal or targeted application. For instance, if the system needs to build a map of the environment after being driven through it, using an off-line method is enough. On the other hand, on-line approaches are required when we want to perform real-time localization of a robot based on an already build map or when doing Simultaneous Localization and Mapping (SLAM).

This article introduces a novel vision based method to detect loop-closure and to build a topological map of the environment. Vision based solutions to complete these tasks usually compute the similarity between camera images, using techniques such as local / global histograms or feature matching.

Techniques exploiting local visual features have a high accuracy and can handle small changes in the image [5]. However, these features by themselves are not particularly discriminative. They are only distinguishable when combining their local descriptor with their global or relative location in the image [6]. Different feature detectors and descriptors have been developed through the years, and some of them are robust to scaling, rotation and change in lighting condition. Among these features, *salient regions* (SR) are known to be quite discriminative, robust to viewpoint change and image perturbations, as well as having a high repeatability rate.

In addition to the previously cited advantages of *salient regions*, we should mention that these features do not rely on shapes or characteristics found in specific environments (like verticality in urban environments, or local planarity in most structured environments). Moreover, the first few regions (on average between 5 to 20) to be detected on an image are quite repeatable. Altogether, this means that our approach based on this type of features and named *SAIL-MAP*, is not limited to a particular kind of environments, and computation time as well as memory requirements are reduced, due to the limited number of regions needed per image.

In Section II, we present a review of related works on visual loop-closure detection. Section III briefly introduces our approach to solve the LCD problem and to build the topological map. The novel visual feature detector and descriptor used in our work are thoroughly presented in Section IV. Section V describes the map representation adopted and provides additional details about the loop-closure detector. Experimental results demonstrating the efficiency of our approach on several data sets are finally given in Section VI. The final section is devoted to the conclusion and future work.

II. RELATED WORKS

Most vision based loop-closure techniques described in the literature differ in the way they detect loop-closure, but share the use of bag-of-words models (BoW) [1], [2], [3], [7], [4] or inverted files [3] to perform efficiently. BoW originated from natural language processing but have been transposed in computer vision, where documents (places or images) are represented by a set of visual words (features). More specifically, visual words are defined by a generative model in [1] [2]. Word histograms are used for loop-closure in [7] and a basically similar but much more efficient approach is presented in [4] using BRIEF descriptors. The

[1]Pascal Institute - UMR 6602 UBP/CNRS - Campus des Cézeaux, 24 Avenue des Landais, 63177 AUBIERE Cedex France, FirstName.LastName@univ-bpclermont.fr

approach presented in [1], [2], known as FAB-MAP (for Fast Appearance Based Mapping), detects loop-closure from omnidirectional camera images and achieves 100% precision, with a recall of respectively 48% and only 3% on 70km and 1000km trajectories. FAB-MAP has become the gold standard for loop-closure detection, but its robustness drops when many images contain very similar structures.

Indeed, these approaches rely on an unstructured set of visual words, and therefore do not capture the geometric information encoded between but not directly among the extracted features. While they allow a very efficient matching, they suffer from perceptual aliasing, because the extracted features are arbitrarily reorganized and forced to match the visual words [8]. While keeping local features that are not necessarily discriminative, a strong descriptor is needed to choose among them. For instance, one of the most simple descriptors would be a vector consisting of the intensity values of pixels surrounding the detected point of interest. An evaluation of more powerful descriptors can be found in [9]. In [1], [2], [3] the features used are SIFT [18] or SURF [19]. These are the most popular ones in this context because they are invariant to scale, rotation as well as lighting change, and behave correctly for slight perspective changes.

In these approaches as well as in ours, we can assume a dense topological map has to be built, where each novel image is treated as a node. Fig. 1 provides an overview of the common framework used for vision based topological map building. As previously explained, the main difference between the approaches lies in the feature detection/description module. In this work, and instead of using classical descriptors, we present a novel descriptor called *attentional descriptor*, obtained at a very limited computational cost from the feature maps (for instance compared to SIFT), that have already been computed by the associated saliency-based detector (see Section IV).

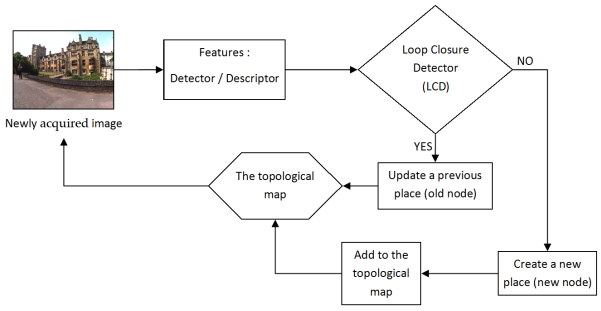

Fig. 1. A global modular view of our topological mapping framework.

III. METHOD OVERVIEW

Fig. 1 depicts a global overview of our framework. Given a query image or newly acquired image, local image features based on saliency are extracted using a dedicated detector and descriptor. The features extracted from the query image are then used for a loop-closure check. Basically, the loop-closure module evaluates the similarity of the query image to all the images attached to the nodes of the topological map. When a loop-closure is detected, which means that the query image is probably acquired at a previously visited place/node in the map. However, if loop-closure does not occur, a new place/node is created and added to the map by connecting it to previously explored nodes.

The following sections present the visual features used in our approach, how the environment is represented within the system and how the map is updated when processing a new image.

IV. VISUAL ATTENTION - SALIENCY REGIONS

We identify salient regions (*SR*) as those regions of an image that are visually more conspicuous because of their contrast with their surroundings. The mechanism of visual saliency has been observed and studied in humans for a number of elementary dimensions of visual stimuli, including color, orientation, depth, motion, and more. In computer vision, several computational models were proposed to reproduce this behavior. Among these models, the most popular is probably the bottom-up saliency algorithm from [14], which defines the contrast as center-surround differences for several elementary dimensions and at any point of the visual stimulus. In this section, the feature detector of the most salient regions is first introduced, followed by the descriptor for each *SR*.

A. Attentional Detector

We use a detector mimicking the attentional system that basically works in a bottom-up fashion, the saliency computations being entirely based on the content of the current image. Computations of the intensity, orientation, and color dimensions are performed on different scales with image pyramids. In contrast to most other attentional models [14], [15], on-off and off-on contrasts for intensity are computed separately in our system. After summing up the single-scale maps into one multi resolution map, this yields 2 intensity maps. Similarly, 4 multi-scale color maps (green, blue, red, yellow) which highlight salient regions for the corresponding colors are produced, and 4 orientation maps $(0°, 45°, 90°, 135°)$ are computed using Gabor filters (for additional details, please refer to [14]). Before any combination of the maps is performed, they are normalized, in order to take into account the uniqueness of the detected feature : a feature which appears seldom in a scene is assigned a higher saliency than a frequently present one. The normalized version $\tilde{X}$ of any given map X is defined by :

$$\tilde{X} = \frac{1}{max(X) \times \sqrt{m}} \times X \qquad (1)$$

where m is the number of local maxima that has a value higher than $max(X/2)$. The normalized maps $\tilde{X}_i$ are then grouped by type of elementary dimensions, and summed into 3 *conspicuity maps* (C_X) : C_I (intensity), C_O (orientation) and C_C (color). There are normalized again and finally merged into a single saliency map S :

$$S = \tilde{C}_I + \tilde{C}_O + \tilde{C}_C \tag{2}$$

First, we select the most salient region (i.e. the one with the highest S value, named MSR) Afterward, the regions containing pixels with a saliency value S higher than $MSR/4$ are extracted as salient regions (SR). To ease storing and processing, SRs are defined as rectangular areas of size 10×10 pixels around the most salient one. Fig. 2 shows the *feature maps* and *conspicuity maps* for one sample image. The salient regions are extracted through an iterative process. For each cycle, we select salient region with the highest saliency value. Then, this saliency in this region is suppressed according to a mechanism inspired from human psychophysics called inhibition of return (IOR), allowing the next SR that has a saliency greater than $MSR/4$ to be found, and so on.

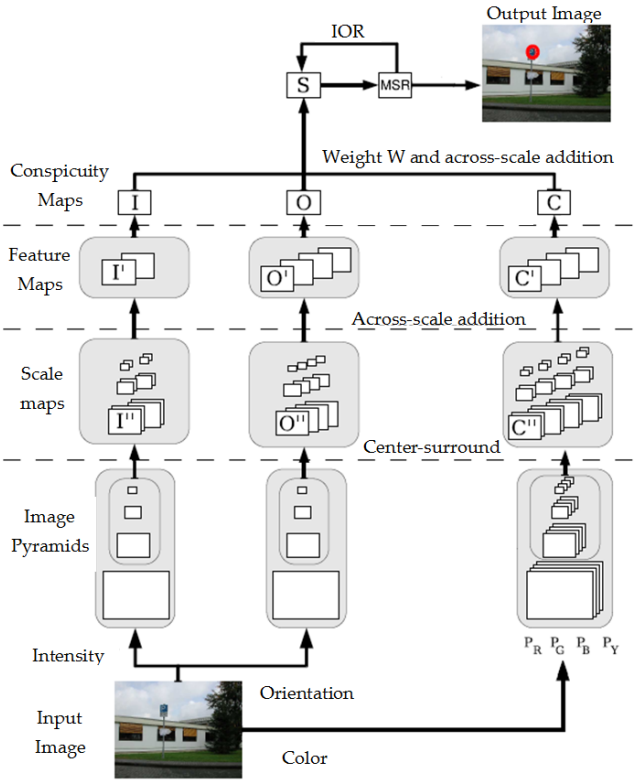

Fig. 2. The bottom-up attentional system. Saliencies to the elementary dimensions (intensity, orientation and color) are computed independently and then combined (from [16]).

B. Attentional Descriptor

For each SR, a feature vector V_{SR} with 10 entries is defined, which describes how much each feature contributes to its associated conspicuity map. We here define the vector component v_i as the ratio of the mean saliency in SR for the feature map X_i (noted $\overline{X_i(SR)}$) and the mean saliency for the corresponding conspicuity map CX_i :

$$V_{SR} = \{v_i\}_{i \in [\![1,10]\!]} \tag{3}$$

Where :

$$v_i = \frac{\overline{X_i(SR)}}{\overline{CX_i(SR)}} \tag{4}$$

A similar descriptor was found in the literature [16], except that the vector descriptor has 13 elements recapitulating the values of the 10 features and the 3 conspicuity maps of the VOCUS model [16]. The value v_i for map X_i is there defined as the ratio of the mean saliency in the salient region to the mean background saliency ($\neg SR$ stands for the entire image but SR) :

$$v_i = \frac{\overline{X_i(SR)}}{\overline{X_i(\neg SR)}} \tag{5}$$

This descriptor was mainly used for tracking applications where the regions to be matched belong to successive camera images, for which the background cannot change a lot (with standard framerates). On the contrary, our goal is to use SRs for loop-closure detection, which means that we need to match images from the same scene taken at very different times, so that the background may be significatively altered (sky, presence or absence of objects...), as illustrated on Fig. 3. Such changes would have a drastic impact on the descriptor, if computed using Eq. 5.

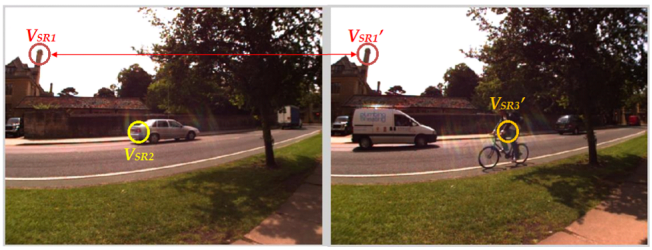

Fig. 3. Two images from the Oxford University data sets took at the same place with different objects on it. A feature vector computed near the chimney with our approach will not be influenced by changes in remote objects, given a high enough resolution.

When detecting the salient regions on the image, we also keep the (x,y) center coordinates of each SR to be able to use geometrical constraints which may consistently improve the results. For each image from which we extract n different SR, we thus define the global descriptor of the image as :

$$V_{image} = \{x_k, y_k, V_{SR_k}\}_{k \in [\![1,n]\!]} \tag{6}$$

Nevertheless, we cannot guarantee that the order in which SR are extracted will be fixed, mainly because of relative differences in saliency, and again due to global and important changes in the image. The components of V_{image} are therefore sorted in descending order of saliency.

Finally, and to make this descriptor even more robust, we extended the matching process of the different SRs with two geometrical constraints that are further described in Section V-B.

V. MAP REPRESENTATION & LOOP-CLOSURE

A. Map representation

A topological map represents the environment as a set of places (represented as nodes of a graph) which can be connected by edges. Edges represent some sort of connectivity between the places. An important and advantageous property of topological maps is their independence to geometrical information about the environment. A traditional example of a topological map is a metro map in which different stations are represented as nodes, where edges connecting pairs of nodes indicate traversability across them. Fig. 4 provides an example of a topological map created in an indoor environment.

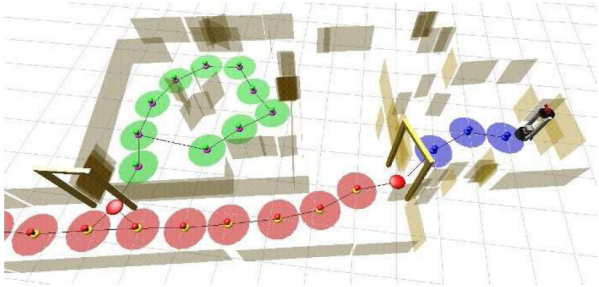

Fig. 4. An example topological map built by a robot.[17]

In our work the built topological map is a dense one, in which every acquired image stands as a node in the topological graph. Each node of this map contain the V_{image} descriptor.

B. Image similarity

Image similarity is estimated using the attentional descriptors of the different images and two geometrical constraints. Let us consider two images I_1 and I_2, from which we have respectively extracted n_1 and n_2 *SR*s, and suppose we want to determine whether they represent the same place. *SR*s that are matched are those that minimize the Euclidian distance between the vectors representing the region descriptors ($d_{ij} = ||V_{SR_i} - V_{SR_j}||_2$). But beforehand, the cost of testing all couples of regions between images ($n_1 \times n_2$) can be alleviated by only considering the regions that validate the two following constraints :

1) Two matched regions should not be far from each other, which means that for any SR_1 and SR_2 respectively in I_1 and I_2, we should only compute d_{12} if $||(x_1,y_1)-(x_2,y_2)||_2 < \Delta$ (see Fig. 5 where the number of distances d to compute is brought down to only 2).
2) The relative position of the different regions in the first image should be approximately the same in the second image (see Fig. 6).

These two constraints allow us to associate each region from I_1 with a reduced number of regions from I_2 ($\ll n_2$). If at least one region is found in the research zone (Fig. 5), we can then proceed to computing the Euclidian distance d and finding the minimum. We can then measure the similarity between the two images using the following formula :

$$\frac{1}{Sim(V_{I_1}, V_{I_2})} = \frac{1}{|MR|} \times \sum_{(i,j)\in MR} d_{ij} \tag{7}$$

where MR is the set of matched region couples, so that SR_i and SR_j are respectively extracted from I_1 and I_2. A matching is more accurate and the similarity is higher when the distances are lower and more regions are matched.

Finally, $Sim(I_1, I_2)$ is compared to a predefined threshold Th_{LCD}, chosen in adequation with the task and the precision required [1]. Thus, if $Sim(I_1, I_2)$ is higher than Th_{LCD}, the two images represent the same place, if not, the second image is considered a new place.

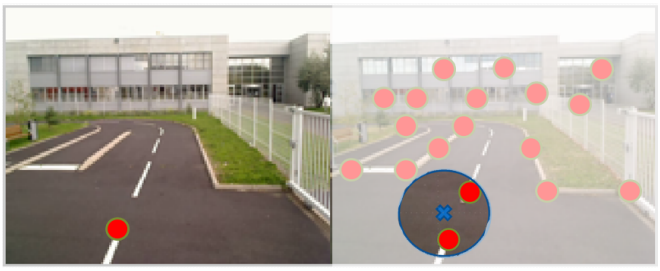

Fig. 5. Search area (Images from PAVIN data sets).

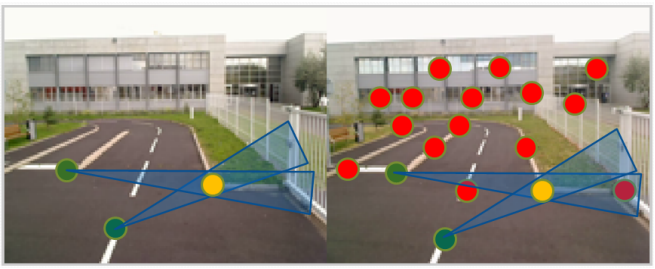

Fig. 6. Relative position of SRs are similar (Images from PAVIN data sets).

C. Loop-closure & Map update

Loop-closure is done by matching the *SRs* from the current image with the *SRs* that are on the different nodes of the topological map. When the robot explores an area or the system is provided a sequence of images, the descriptor of every newly processed image is compared to the descriptors stored in the nodes already present in the map to detect a potential loop-closure. This means that, the similarity (Eq. 7) is computed between the current image descriptor and all the descriptors attached to the nodes. After computing all the similarities, we search for the maximum and compare its value to Th_{LCD}. If it is higher than Th_{LCD}, the system simply memorizes the corresponding node (set as active). If

1. In our experiment $Th_{LCD} = 20$ was empirically chosen to get the results presented in section VI

it is lower than Th_{LCD}, a new node is added to the map with the descriptor associated with the current image V_I, a new edge is created to connect it to the previously activated node, and the new node is finally set as active (see Figure-4). This is true but for the very first image, for which the descriptor is automatically added to the map as its first node.

VI. EXPERIMENTAL RESULTS

We compare our results to the results obtained with the algorithm presented in [1] known as FAB-MAP. It makes use of SIFT features and the bag-of-words technique. Loop-closure accuracies are measured by means of precision/recall values. Precision is defined as the ratio of true positive loop-closure detections to total detections, and recall is the ratio of true positive loop-closure detections to the number of ground truth loop-closure.

The test was done on different data sets, two from the Oxford university (UK) and two from Institut Pascal (France). Fig-7 shows a top view using Google-Map of the four locations (New College - City Center - PAVIN - CEZEAUX).

A. Data sets

The Oxford university data sets were collected using a mobile robot on an outdoor environment. As the robot travels through the environment, it collects images to the left and right of its trajectory approximately every 1.5 m. The first data set *New College* **(NC)** was chosen to test the system's robustness to perceptual aliasing. It features several large areas of strong visual repetition, including a medieval cloister with identical repeating archways and a garden area with a long stretch of uniform stone wall and bushes. The second data set labeled *City Center* **(CC)** was chosen to test matching ability in the presence of scene change. It was collected along public roads near the city center, and features many dynamic objects such as traffic and pedestrians. Additionally, it was collected on a windy day with bright sunshine, which makes the abundant foliage and shadow features unstable.

In the Institut Pascal data sets two image sequences are used. Both are freely available data sets[2]. The data sets are acquired using a VIPALAB platform. This is a car-like vehicle designed to serve as a prototype for research and development in autonomous urban transport vehicles. The color images are acquired by a Logitech QuickCam Pro 9000 webcam embedded on the roof of the vehicle and looking forwards. There is also a GPS system that provides an absolute localization measurements, which serves only as ground-truth for loop-closure and localization validation. The two sequences PAVIN **(PV)** and CEZEAUX **(CZ)** are named so after the places where they were acquired. PAVIN is an artificial village which simulates urban and rural environments with tarred roads, rural-style roads, road markers, traffic signals, roundabouts and junctions. CEZEAUX is the area surrounding Institut Pascal and Universite Blaise Pascal and is a semi-urban style environment with roads, buildings, lawns and vegetation on either sides of the road. More details about these sequences can be found in the web site : `http://ipds.univ-bpclermont.fr/`. For each place two different paths were taken called (Heman and Faust) for PAVIN, and (Heko and Sealiz) for CEZEAUX.

2. Data sets acquisition supported by our laboratory : Pascal Institute. Hosted at `http://ipds.univ-bpclermont.fr/`

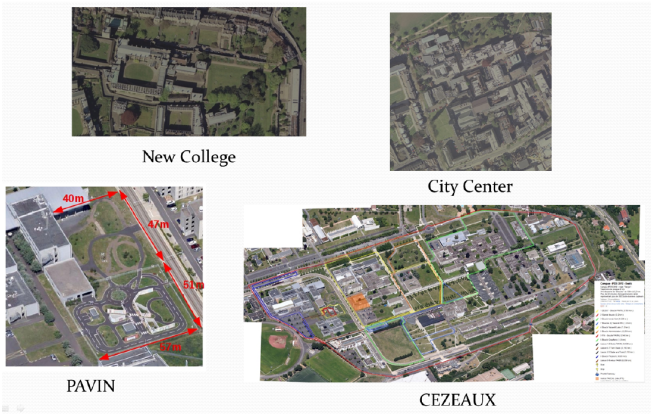

Fig. 7. The four locations where the data sets were acquired (images from Google-Map).

B. The Results

Table-I gives an overview of the results obtained when using the FAB-MAP and the SAIL-MAP algorithms respectively on the different data sets. The results of precision and recall are also shown in Fig 8.

		Precision (%)		Recall(%)	
Data Sets	Length	FAB-MAP	SAIL-MAP	FAB-MAP	SAIL-MAP
CC-Left	2.0 km	**71**	64	52	**53**
CC-Right	2.0 km	**86**	72	**54**	43
NC-Left	1.9 km	**91**	73	46	**62**
NC-Right	1.9 km	90	**95**	66	**71**
PV-Heman	0.6 km	**95**	90	43	**75**
PV-Faust	1.3 km	**100**	98	53	**91**
CZ-Heko	4.2 km	**97**	94	47	**50**
CZ-Sealiz	7.8 km	96	**98**	40	**67**

TABLE I
THE PRECISION AND RECALL SCORES USING FAB-MAP 2.0 AND SAIL-MAP ON THE DIFFERENT DATA SETS.

From the Fig-8 we can see that the SAIL-MAP algorithm offers higher scores for recall comparing to the FAB-MAP except for one data set which is the *CC-Right*. The average difference in recall across the different data sets is $\simeq 16\%$. With the PV-Faust data set recall using SAIL-MAP was equal to 91% whereas with FAB-MAP it was equal to 53%, which shows that SAIL-MAP was more able to detect the real loop-closure, and even if with FAB-MAP we obtained 100% for the precision, a 98% was obtained using SAIL-MAP, which is also a good results. In the other hand as shown in Fig-8 when talking about precision SAIL-MAP results are close to the one obtained using FAB-MAP, and the average difference across the data sets is $\simeq 7\%$. It should also be mentioned that whenever FAB-MAP gives a score higher than 90% for

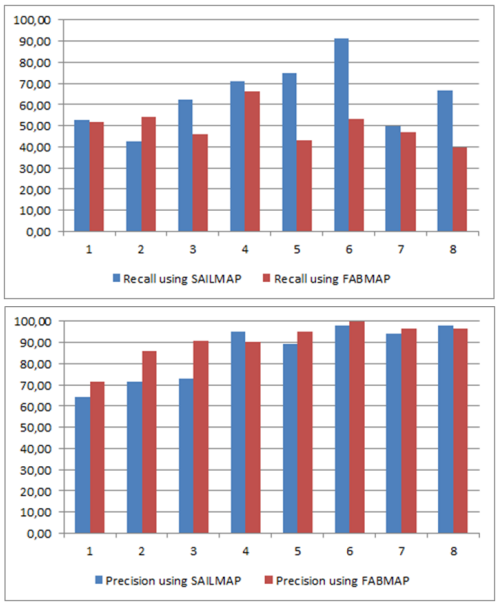

Fig. 8. (Top) The RECALL scores. (Bottom) The PRECISION scores. The data sets : 1-CC-Left, 2-CC-Right, 3-NC-Left, 4-NC-Right, 5-PV-Heman, 6-PV-Faust, 7-CZ-Heko, 8-CZ-Sealiz.

the precision, SAIL-MAP is close to it, except for the NC-Left where the FAB-MAP offered 91% precision against 73% with SAIL-MAP.

VII. CONCLUSION AND FUTURE WORK

The approach SAIL-MAP proposed in this article to build dense topological map and to resolve the LCD problem, can be also used in future work to build sparse topological maps in which each node represents a group of images rather than representing individual image.

The descriptor proposed is not influenced by the different changes that could occurs in the background scene, because each element of the descriptor is computed locally. It should also be noted that the computations are common to the detector and the descriptor.

The different results provided in this article were done in simulation mode using Matlab software, the next step in our research is to test this architecture in realtime using the VIPALAB platform and the C++ programming language. If the realtime processing is not reached with such a programming language, the solution that we can propose knowing that salient regions can be extracted by an intrinsically and massively parallel algorithm is the use of a dedicated hardware such as FPGA.

REFERENCES

[1] Mark Joseph Cummins and Paul M. Newman. Fab-map : Probabilistic localization and mapping in the space of appearance. I. J. Robotic Res, pages 647-665, 2008.

[2] Mark Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In Jek Trinkle, Yoky Matsuoka, and Jos A. Castellanos, editors, Robotics : Science and Systems. The MIT Press, 2009.

[3] 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, vol. 24(5), pp. 1024-1037, Oct. 2008.

[4] D. Galvez-Lopez and J. TArdos, Real-time loop detection with bags of binary words, in IEEE/RSJ International Confercen on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, Sep.2011,pp 51-58.

[5] Hong Zhang, Bo Li, and Dan Yang. Keyframe detection for appearance-based visual slam. In IROS, pages 2071-2076. IEEE, 2010.

[6] Kin Leong Ho and Paul Newman. Detecting loop-closure with scene sequences. Int. J. Comput. Vision, pages 261-286, September 2007.

[7] F.Fraundorfer, C.Engels, and D. Nister, Topological mapping localization and navigation using image collections, in IEEE/RSJ Internationam Conference on Intelligent Robots and Systems, IROS07.

[8] Hossein Shahbazi and Hong Zhang. Application of locality sensitive hashing to realtime loop-closure detection. In IROS, pages 1228-1233. IEEE, 2011.

[9] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell, pages 1615-1630, October 2005.

[10] Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. The International Journal of Robotics Research, pages 735-758, August 2002.

[11] L. Goncalves, E. Di Bernardo, D. Benson, M. Svedman, J. Ostrowski, N. Karlsson, and P. Pirjanian. A visual front-end for simultaneous localization and mapping. In 2005 IEEE International Conf. on Robotics and Automation, ICRA.

[12] Patric Jensfelt, Danica Kragic, John Folkesson, and Mrten Bjrkman. A frame work for vision based bearing only 3d slam. In ICRA, pages 1944-1950. IEEE, 2006.

[13] Mark Cummins and Paul M. Newman. Probabilistic appearance based navigation and loop closing. In ICRA, pages 2042-2048.IEEE, 2007.

[14] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell, pages 1254-1259, November 1998.

[15] N. Ouerhani, A. Bur, and H Hugli. Visual attention-based robot self-localization. in Proc. of Europ. Conf. on Mobile Robotics (ECMR), 2005.

[16] Simone Frintrop. VOCUS : A Visual Attention System for Object Detection and Goal Directed Search, volume 3899 of Lecture Notesin Computer Science. Springer, 2006.

[17] Hemanth Korrapati, Loop-Closure for Topological Mapping and Navigation with Omnidirectional Images, 2013, Thesis, Blaise Pascal University, Clermont-Ferrand, France

[18] D. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[19] H. Bay, T. Tuytelaars, and L. Van Gool, SURF : Speeded up robust features, Computer Vision–ECCV 2006, pp. 404–417, 2006.