Memory Management for Real-Time Appearance-Based Loop Closure Detection

Mathieu Labbé and François Michaud
Department of Electrical and Computer Engineering
Université de Sherbrooke, Sherbrooke, QC, CA J1K 2R1
{mathieu.m.labbe, francois.michaud}@usherbrooke.ca

Abstract— **Loop closure detection is the process involved when trying to find a match between the current and a previously visited locations in SLAM. Over time, the amount of time required to process new observations increases with the size of the internal map, which may influence real-time processing. In this paper, we present a novel real-time loop closure detection approach for large-scale and long-term SLAM. Our approach is based on a memory management method that keeps computation time for each new observation under a fixed limit. Results demonstrate the approach's adaptability and scalability using four standard data sets.**

I. INTRODUCTION

Autonomous robots operating in real life settings must be able to navigate in large, unstructured, dynamic and unknown spaces. Simultaneous localization and mapping (SLAM) [1] is the capability required by robots to build and update a map of its operating environment and to localize itself in it. A key feature in SLAM is to recognize previously visited locations. This process is also known as loop closure detection, referring to the fact that coming back to a previously visited location makes it possible to associate this location with another one recently visited.

For global loop closure detection approaches [2]–[6], a general way to detect a loop closure is to compare a new location with the ones associated with previously visited locations. If no match is found, then a new location is added. A problem with this approach is that the amount of time required to process new observations increases with the number of locations in the map. If this becomes larger than the acquisition time, a delay is introduced, leading to an obsolete map. In addition, a robot operating in large areas for a long period of time will ultimately build a very large map, making updating and processing the map difficult to achieve in real-time.

Our solution is to manage the locations used to detect loop closures based on their occurrences. The idea is simple (inspired from works in psychology [7], [8]): it consists of keeping the most recent and frequently observed locations in the robot's Working Memory (WM), and transferring the others into a Long-Term Memory (LTM). When a match is found between the current location and one stored in WM, associated locations stored in LTM can be remembered and updated.

This paper describes our memory management approach to accomplish appearance-based loop closure detection with real-time constraint for long-term SLAM. Processing time is the criterion used to limit the number of locations kept in WM, and therefore satisfy real-time constraint. Our memory management mechanism ensures satisfaction of real-time constraint independently of the scale of the mapped environment. For this reason, we named our approach Real-Time Appearance-Based Mapping (RTAB-Map). Trials conducted with four standard data sets demonstrate real-time capabilities of our approach under various conditions.

The paper is organized as follows. Section II summarizes related work on loop closure detection for large-scale SLAM. Section III describes our approach. Section IV presents experimental results and Section V shows a qualitative comparison of our approach with similar ones in the community. Section VI concludes the paper.

II. RELATED WORK

For most of the probabilistic SLAM approaches [1], [9]–[14], loop closure detection is done locally, i.e., matches are found between new observations and a limited region of the map, so that process can be accomplished in real-time and at 30 Hz [15]. The size of the region is determined by the uncertainty associated with the robot's position. Such approaches fail if the estimated position is erroneous. Newman et al. [2] indicate that in real world situations, it is very likely that events will cause gross errors in the estimated position. Therefore, it is preferable to do loop closure detection without using an estimated position.

Global loop closure detection consists of comparing a new observation to observations stored in the map, and to add this observation only if no match is found. Vision is the sense generally used because of the distinctiveness of the features extracted from the environment [16]–[18], although successful large-scale mapping using features extracted using laser range finder data is possible [19]. For vision-based mapping, the bag-of-words [20] approach is commonly used [3], [5], [6], [21] and has shown to perform loop closure detection in real-time for paths of up to 1000 km [22]. The bag-of-words approach consists of representing each image by visual words taken from a dictionary. The visual words are usually made from local feature descriptors, such as Speeded-Up Robust Features (SURF) [23]. Each word keeps a link to images it is associated to, making image retrieval efficient over large data set. However, as a robot explores new areas over long periods of time, the number of images to compare increases linearly, making it eventually impossible to conduct the matching process in real-time. The number of comparisons can be decreased by considering only a selection of previously acquired images (referred to as key images) for the matching process, while keeping detection performance nearly the same to using all images [24]. However, complexity still increases linearly with the number of key images. This phenomenon can also occur in close and dynamic environments in which the robot frequently visits the same locations: perceptual aliasing, changes that can occur in dynamic environments or the lack of discriminative information may affect the ability to recognize previously visited locations, leading to the addition of new locations in the map and consequently affecting the satisfaction of real-time constraints [25]. To limit the growth of the number of images to match, pruning [26] or clustering [27] methods can be used to consolidate portions of the map which exceed a spatial density threshold. This limits growth over time, but not according to the size of the explored environment.

III. REAL-TIME APPEARANCE-BASED MAPPING (RTAB-MAP)

The objective of our work is to provide a solution independent of time and size, to achieve real-time loop closure detection for long-term operation in large environments. The idea resides in only using a certain number of locations for loop closure detection so that real-time constraint can be satisfied, while still having access to the locations of the entire map when necessary. When the number of locations in the map makes processing cycle time for finding matches greater than a real-time threshold, our approach transfers locations less likely to cause loop closure detection from the robot's WM to LTM, so that they do not take part in the detection of loop closures. However, if a loop closure is detected, neighbor locations can be retrieved and brought back into WM to be considered in loop closure detections. Note that our approach can be seen as a dynamic way to generate and to dynamically maintain a constant number of key locations in WM, which differs from other key frame approaches [24] where the number of key locations would increase over time and space.

Transferred locations in LTM are not used to find a match with the current one. Therefore, some loops may not be detected, making it important to carefully choose which locations to transfer to LTM. A naive approach is to use a first-in first-out (FIFO) policy, pruning the oldest locations from the map. However, this sets a limit on the maximum number of locations memorized in sequence when exploring an environment: if the real-time limit is reached before loop closure can be detected, pruning the older locations will make it impossible to eventually find a match. Another approach would be to randomly pick locations to transfer, but it is preferable to keep in WM the locations that are more susceptible to be revisited. Therefore, our proposed approach evaluates the number of times a location has been matched or consecutively viewed (referred to as Rehearsal) to set its weight, making it possible to identify the locations seen more frequently than others and that are more likely to cause loop closure detections. When transfer occurs, the location with the lowest weight is selected. If there are locations with the same weight, then the oldest one is selected to be transferred.

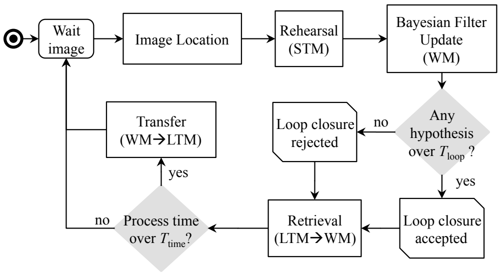

Fig. 1. Flow chart of our memory management loop closure detection processing cycle.

Similar to [4], locations in the Short-Term Memory (STM) are not used for loop closure detection, to avoid loop closure detection on locations that have just been visited (the last location always looks similar to the most recent ones). STM size is fixed based on the robot velocity and the rate at which the locations are acquired. When the number of locations reaches STM size limit, the oldest location in STM is moved into WM.

Fig. 1 illustrates the overall loop closure detection process which is explained in details in the following sections.

A. *Image Location*

The bag-of-words approach [20] is used to create a signature z_t of an image acquired at time t. An image signature is represented by a set of visual words contained in a visual dictionary incrementally constructed online using a randomized forest of kd-trees [28]. Speeded-Up Robust Features (SURF) [23] are extracted from the image as the visual words. A location L_t is then created with signature z_t, a weight initialized to 0 and a bidirectional link in the graph with L_{t-1}. Note that the dictionary contains only words of locations in WM and STM.

B. *Rehearsal*

To update the weight of the acquired location, L_t is compared to the ones in STM from the most recent ones to the latest, and similarity $\mathfrak{s}$ is evaluated using (1) :

$$\mathfrak{s}(z_t, z_c) = \begin{cases} N_{\text{pair}}/N_{z_t}, & \text{if } N_{z_t} \geq N_{z_c} \\ N_{\text{pair}}/N_{z_c}, & \text{if } N_{z_t} < N_{z_c} \end{cases} \quad (1)$$

where N_{pair} is the number of matched word pairs between the compared location signatures, and N_{z_t} and N_{z_c} are the total number of words of signature z_t and the compared signature z_c respectively. If $\mathfrak{s}(z_t, z_c)$ is higher than a fixed similarity threshold $T_{\text{rehearsal}}$, L_c is merged into L_t and Rehearsal stops. In Rehearsal, only words of z_c are kept in the merged signature : z_t is cleared and z_c is copied into z_t. To complete the merging process, the weight of L_t is increased by the one of L_c plus one, the neighbor links of L_c are added to L_t and L_c is deleted from STM.

C. Bayesian Filter Update

The role of the discrete Bayesian filter is to keep track of loop closure hypotheses by estimating the probability that the current location L_t matches one of an already visited location stored in the WM. Let S_t be a random variable representing the states of all loop closure hypotheses at time t. $S_t = i$ is the probability that L_t closes a loop with a past location L_i, thus detecting that L_t and L_i represent the same location. $S_t = -1$ is the probability that L_t is a new location. The filter estimates the full posterior probability $\boldsymbol{p}(S_t|L^t)$ for all $i = -1, ..., t_n$, where t_n is the time index associated with the newest location in the WM, expressed as follows [21]:

$$\boldsymbol{p}(S_t|L^t) = \eta\, \underbrace{\boldsymbol{p}(L_t|S_t)}_{\text{Observation}} \underbrace{\sum_{i=-1}^{t_n} \underbrace{\boldsymbol{p}(S_t|S_{t-1}=i)}_{\text{Transition}} p(S_{t-1}=i|L^{t-1})}_{\text{Belief}} \tag{2}$$

where η is a normalization term and $L^t = L_{-1}, ..., L_t$. Note that the sequence of locations L^t includes only the locations contained in WM and STM, thus L^t changes over time as new locations are created and when some locations are retrieved from LTM or transferred to LTM, deviating from the classical Bayesian filtering where such sequences are fixed.

The observation model $\boldsymbol{p}(L_t|S_t)$ is evaluated using a likelihood function $\mathfrak{L}(S_t|L_t)$: the current location L_t is compared using (1) to locations corresponding to each loop closure state $S_t = j$ where $j = 0, .., t_n$, giving a score $s_j = \mathfrak{s}(z_t, z_j)$. Each score is then normalized using the difference between the score s_j and the standard deviation σ, normalized by the mean μ of all non-null scores, as in (3) [21] :

$$p(L_t|S_t = j) = \mathfrak{L}(S_t = j|L_t) = \begin{cases} \frac{s_j - \sigma}{\mu}, & \text{if } s_j \geq \mu + \sigma \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

For the new location probability $S_t = -1$, the likelihood is evaluated using (4) :

$$p(L_t|S_t = -1) = \mathfrak{L}(S_t = -1|L_t) = \frac{\mu}{\sigma} + 1 \tag{4}$$

where the score is relative to μ on σ ratio. If $\mathfrak{L}(S_t = -1|L_t)$ is high, meaning that L_t is not similar to a particular one in WM (when $\sigma < \mu$), then L_t is more likely to be a new location.

The transition model $\boldsymbol{p}(S_t|S_{t-1} = i)$ is used to predict the distribution of S_t, given each state of the distribution S_{t-1} in accordance with the robot's motion between t and $t-1$. Combined with $p(S_{t-1} = i|L^{t-1})$ (i.e., the recursive part of the filter) this constitutes the belief of the next loop closure. The transition model is expressed as in [21].

1) $p(S_t = -1|S_{t-1} = -1) = 0.9$, the probability of a new location event at time t given that no loop closure occurred at time $t-1$.
2) $p(S_t = i|S_{t-1} = -1) = 0.1/N_{\text{WM}}$ with $i \in [0; t_n]$, the probability of a loop closure event at time t given that no loop closure occurred at $t-1$. N_{WM} is the number of locations in WM of the current iteration.
3) $p(S_t = -1|S_{t-1} = j) = 0.1$ with $j \in [0; t_n]$, the probability of a new location event at time t given that a loop closure occurred at time $t-1$ with j.
4) $p(S_t = i|S_{t-1} = j)$, with $i, j \in [0; t_n]$, the probability of a loop closure event at time t given that a loop closure occured at time $t-1$ on a neighbor location. The probability is defined as a discretized Gaussian curve centered on j and where values are non-null for exactly eight neighbors (for $i = j-4, ..., j+4$) and their sum are 0.9.

D. Loop Closure Hypothesis Selection

When $\boldsymbol{p}(S_t|L^t)$ has been updated and normalized, the highest hypothesis of $\boldsymbol{p}(S_t|L^t)$ greater than the loop closure threshold T_{loop} is selected. Because a loop closure hypothesis is generally diffused to its neighbors (neighbor locations share some similar words between them), for each probability $p(S_t = i|L_t)$ where $i \geq 0$, the probabilities of its neighbors in the range defined by the transition model $p(S_t = i|S_{t-1} = j)$ are summed together. Note that for a selection to occur, a minimum of T_{minHyp} locations must be in WM: when the graph is initialized, the first locations added to WM have automatically a high loop closure probability (after normalization by (2)), therefore wrong loop closures could be accepted if the hypotheses are over T_{loop}.

If there is a loop closure hypothesis $p(S_t = i|L^t)$ where $i \geq 0$ is over T_{loop}, the hypothesis $S_t = i$ is then accepted, L_t is merged with the old location L_i: the weight of L_t is increased by the one of L_i plus one and the neighbor links of L_i are added to L_t. After updating L_t, unlike Rehearsal, L_i is not immediately deleted from the memory. For the constancy of the Bayesian filter with the next acquired images, the associated loop closure hypothesis S_i must be evaluated until L_i is not anymore in the neighborhood of the highest loop closure hypothesis, after which it is deleted.

E. Retrieval

Neighbors of location L_i in LTM are transferred back into WM if $p(S_t = i|L^t)$ is the highest probability. Because this step is time consuming, a maximum of two locations are retrieved at each iteration (chosen inside the neighboring range defined in Section III-C). The visual dictionary is updated with the words associated with the signatures of the retrieved locations. Common words from the retrieved signatures still exist in the dictionary, so a simple reference is added between these words and the corresponding signatures. For words that are not present anymore (because they were removed from the dictionary when the locations were transferred), they are matched to those in the dictionary to find if more recent words represent the same SURF features. This step is particularly important because the new words added from the new signature z_t may be identical to the previously transferred words. For matched words, the old words are replaced by the new ones in the retrieved signatures. All the references in the LTM are changed and the old words are permanently removed. If some words are still unmatched, they are simply added to the dictionary.

F. Transfer

When processing time becomes greater than T_{time}, the oldest location with the lowest weight is transferred from WM to LTM. T_{time} must be set to allow the robot to process the perceived images in real-time. Higher T_{time} means that more locations (and implicitly more words) can be kept in the WM, and more loop closure hypotheses can be kept to better represent the overall environment. T_{time} must therefore be set according to the robot's CPU capabilities, computational load and operating environment. If T_{time} is set to be higher than the image acquisition time, the algorithm intrinsically uses an image rate corresponding to T_{time}, with 100% CPU usage. As a rule of thumb, T_{time} can be set to about 200 to 400 ms smaller to the image acquisition rate at 1 Hz, to ensure that all images are processed under the image acquisition rate, even if the processing time goes over T_{time} (T_{time} then corresponds to the average processing time of an image by our algorithm). So, for an image acquisition rate of 1 Hz, T_{time} could be set between 600 ms to 800 ms.

Because the most expensive step of our algorithm is to rebuild the visual dictionary, process time can be regulated by changing the dictionary size, which indirectly influences the WM size. A signature of a location transferred to LTM removes its word references from the visual dictionary. If a word does not have reference to a signature anymore, it is transferred into LTM. While the number of words transferred from the dictionary is less than the number of words added from the new or retrieved locations, more locations are transferred. At the end of this process, the dictionary size is smaller than before the new words from the new and retrieved locations were added, thus reducing the time required to update the dictionary for the next acquired image. Saving the transferred locations into the database is done asynchronously using a background thread, leading to a minimal time overhead. Note also that to be able to evaluate appropriately loop closure hypotheses using the discrete Bayesian filter, the retrieved locations of the current iteration are not allowed to be transferred.

IV. RESULTS

Performance of our approach is evaluated in terms of precision-recall metrics [22]. Precision is the ratio of true positive loop closure detections to total detections. Recall is defined as the ratio of true positive loop closure detections to the number of ground truth loop closures. Because the precision is very important in loop closure detection, all results shown in this paper were done at 100% precision, thus no false positive was accepted.

Using a MacBook Pro 2.66 GHz Intel Core i7, we conducted tests with the following community data sets: NewCollege and CityCentre data sets [3], and Lip6Indoor and Lip6Outdoor data sets [21]. The NewCollege and CityCentre data sets contain images acquired from two cameras (left and right). Because our approach only takes one image as input in its current implementation, the two images were merged into one, resulting in one 1280×480 image. With our bounded data sets, we could use the LTM database directly in computer's RAM, but we preferred to use it on the hard drive to simulate a more realistic setup for timing performances.

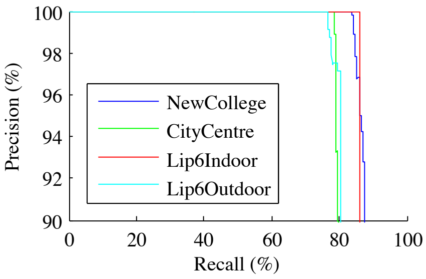

Fig. 2. Precision-recall curves for each data set.

Experimentation of our loop closure detection approach is done using parameters $T_{\text{rehearsal}}$ set to 20%, STM size to 25, T_{loop} to 10% and T_{minHyp} to 15. The same set of parameters is used over the different data sets to evaluate adaptability of our approach without optimization for specific conditions. For this reason, we did not do an exhaustive search for the best parameters; instead, they were set empirically to avoid false positives (precision of 100%) over all data sets while giving good overall recall performances.

For each data set, three experiments were done with different T_{time} to show the effect of memory management on the recall performances. When $T_{\text{time}} = \infty$, all locations are kept in WM, and therefore loop closure detection is done over all previously visited locations, thus giving optimal results. These results are used to compare those when real-time constraint is applied, i.e., with T_{time} set based on the guideline presented in Section III-F ($T_{\text{time}} = 1.4\,\text{s}$ for images acquired at 0.5 Hz, i.e., every 2 seconds; or $T_{\text{time}} = 0.7\,\text{s}$ for images acquired at 1 Hz), and with T_{time} set smaller to such guideline to show the influence of a smaller WM on recall and timing performances. Smaller T_{time} could be used but with lower recall performances if some locations with a high weight are transferred before the robot returns in the area of those locations. Remember also that WM must have a minimum of T_{minHyp} locations to efficiently evaluate loop closure probabilities.

Table I summarizes experimental conditions and results. Results (shown in bold) are the ones done with T_{time} set to values suggested accordingly to image rate. For the NewCollege and CityCentre data sets, RTAB-Map achieves similar performances to when no locations are transferred to LTM ($T_{\text{time}} = \infty$), but respecting real-time constraints. For the Lip6Indoor and Lip6Outdoor, because processing time never reached the real-time constraint, results at $T_{\text{time}} = \infty$ are the same than with the experiments with $T_{\text{time}} = 0.7\,\text{s}$ and $T_{\text{time}} = 1.4\,\text{s}$ respectively. Using a smaller T_{time} does not influence much recall performances, athough in some cases (like with the Lip6Outdoor data set) better results are obtained because some noisy locations, which could incorrectly have high likelihoods, are not in WM, resulting

TABLE I

Experimental conditions and results of RTAB-Map for the community data sets

Data set	NewCollege			CityCentre			Lip6Indoor			Lip6Outdoor		
# images	1073			1237			388			531		
Image rate	≈0.5 Hz			≈0.5 Hz			1 Hz			0.5 Hz		
Image size	1280 × 480			1280 × 480			240 × 192			240 × 192		
T_{time} (s)	∞	**1.4**	0.7	∞	**1.4**	0.7	∞	**0.7**	0.1	∞	**1.4**	0.4
Max WM size (locations)	524	**299**	147	724	**366**	159	115	**115**	80	280	**280**	102
Max dictionary size ($\times 10^3$ words)	169	**99**	52	224	**106**	52	12	**12**	8	89	**89**	31
Max iteration time (s)	2.95	**1.75**	0.95	3.57	**1.83**	1.00	0.185	**0.185**	0.13	1.34	**1.34**	0.51
Precision (%)	100	**100**	100	100	**100**	100	100	**100**	100	100	**100**	100
Recall (%)	85	**84**	85	80	**79**	79	86	**86**	79	77	**77**	81

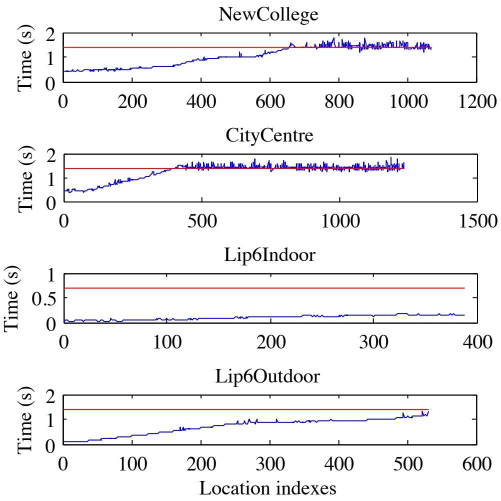

Fig. 3. Processing time in terms of locations processed over time for each data set. The time limit T_{time} used is shown as a red line.

in higher hypotheses on correct previously visited locations. Typical precision-recall curves are shown in Fig. 2, where higher recall performances at 100% precision are the better. Compared to other approaches, RTAB-Map achieves better recall performances at 100% precision than those reported using FAB-MAP in [3] (47% for NewCollege, 37% for CityCentre) and those in [21] (80% for Lip6Indoor, 71% for Lip6Outdoor).

Fig. 3 shows processing time results for each data set. For the NewCollege and CityCentre data sets, the processing time for each new acquired image increases until T_{time} is reached, then it remains around T_{time} and below the real-time limit (set to 0.5 Hz, i.e., 2 seconds). For the Lip6Indoor and Lip6Outdoor data sets, the time taken never reached T_{time}: the smaller slope after processing half of the Lip6Outdoor data set is caused by many loop closures detected, and then memory size increased very slowly.

V. DISCUSSION

Table II compares the proposed loop closure detection approach (RTAB-Map) with Incremental Appearance-Based Mapping (IAB-Map) [4], [21] and FAB-MAP based approaches [3], [22], [29], in terms of: adaptability (the ability of the method to adapt itself to the environment), usability (ease of use in SLAM), repeatability (determinism), hypothesis quality (robustness to perceptual aliasing) and real-time satisfaction (under a maximum of data acquired or not).

RTAB-Map and IAB-Map approaches always have a visual words dictionary which represents well the environment, updating itself while the environment changes. FAB-MAP uses a fixed dictionary, pre-trained with images representing the target environment, required because of its more complex method of hypothesis filtering. The same dictionary could be used over different environments but performances are not optimal [3], limiting adaptability. A pre-computed dictionary is however useful to provide priors on the probability of appearance of a word (using its frequency in the dictionary), giving better likelihood values. While a pre-computed dictionary would give better likelihood values (for its target environment), we preferred to use a incremental one because it is simpler to setup, increasing usability. FAB-MAP has also lower usability because it requires non-overlapping images derived from a pre-filtering (using odometry) done over the acquired images. The two other methods manage overlapping images automatically, discarding (like IAB-Map) or using (like RTAB-Map) them, thus making them easier to use. IAB-Map and FAB-MAP are deterministic, i.e., they lead to the same recall performance when processing the same data set. Because RTAB-Map transfers locations based on computation time, which is affected by other processes running on the robot's computer, a small latency while accessing the hard drive (which plays the role of the database) may cause changes in WM, leading to small differences in performances. This is not a limitation however, because in real life settings the environment is dynamic, causing a similar effect. Robustness to perceptual aliasing is handled in all approaches, but FAB-MAP handles it at the word level rather than at the signature level. Finally, regarding real-time processing for long-term operation, our approach is the only one that is not limited by a maximum of images acquired, a very important criterion in our opinion to achieve long-term SLAM. All approaches are limited by the memory available on the robot, although RTAB-Map is limited by the hard drive size and not the computer's RAM size.

TABLE II

COMPARISON WITH LOOP CLOSURE DETECTION APPROACHES.

Approach	Signature type	Adaptability	Usability	Repeatability	Hypothesis quality	Real-time satisfaction
IAB-Map [21]	SIFT+H histograms	+	-	+	-	limited
IAB-Map [4]	SIFT	+	+	+	-	limited
FAB-Map [3]	SURF	-	-	+	+	limited
Acc. FAB-MAP [29]	SURF	-	-	+	+	limited
FAB-MAP 2.0 [22]	SURF	-	-	+	+	limited
RTAB-Map	SURF	+	+	-	-	unlimited

VI. CONCLUSION

Results presented in this paper suggest that RTAB-Map, a loop closure detection approach based on a memory management mechanism, is able to meet real-time constraints needed for long-term online mapping. While keeping a relatively constant number of locations in WM, real-time processing is achieved for each new image acquired. Retrieval is a key feature that allows our approach to reach adequate recall ratio even when transferring a high proportion of the perceived locations. In future work, we want to test the approach over a larger data set to evaluate more properly the impact when some locations can be seen much more often than others, a situation which could affect the recall ratio over time.

VII. ACKNOWLEDGMENT

F. Michaud holds the Canada Research Chair in Mobile Robotics and Autonomous Intelligent Systems. Support for this work is provided by the Natural Sciences and Engineering Research Council of Canada, the Canada Research Chair program and the Canadian Foundation for Innovation.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[2] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 1180–7.

[3] M. Cummins and P. Newman, "FAB-MAP: probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–65, June 2008.

[4] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat, "Incremental vision-based topological SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1031–1036.

[5] T. Botterill, S. Mills, and R. Green, "Bag-of-words-driven, single-camera simultaneous localization and mapping," *Journal of Field Robotics*, vol. 28, no. 2, pp. 204–226, 2011.

[6] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 941–957, July 2010.

[7] R. Atkinson and R. Shiffrin, "Human memory: A proposed system and its control processes," in *Psychology of Learning and Motivation: Advances in Research and Theory*. Elsevier, 1968, vol. 2, pp. 89–195.

[8] A. Baddeley, *Human Memory: Theory and Practice*. Psychology Pr, 1997.

[9] J. Folkesson and H. I. Christensen, "Closing the loop with graphical SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 731–41, August 2007.

[10] J.-L. Blanco, J. Fernandez-Madrigal, and J. Gonzalez, "Toward a unified Bayesian approach to hybrid metric–topological SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 259–270, April 2008.

[11] J. Callmer, K. Granström, J. Nieto, and F. Ramos, "Tree of words for visual loop closure detection in urban SLAM," in *Proceedings of the Australasian Conference on Robotics and Automation*, 2008, p. 8.

[12] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: EKF SLAM in O(n)," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1107–1120, 2008.

[13] P. Pinies and J. D. Tardós, "Large-scale SLAM building conditionally independent local maps: Application to monocular vision," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1094–1106, October 2008.

[14] D. Schleicher, L. Bergasa, M. Ocaña, R. Barea, and E. López, "Real-time hierarchical stereo Visual SLAM in large-scale environments," *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 991–1002, 2010.

[15] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[16] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, October 2008.

[17] P. Newman and K. Ho, "SLAM- loop closing with visually salient features," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 635–642.

[18] A. Tapus and R. Siegwart, "Topological SLAM," in *Springer Tracts in Advanced Robotics*. Springer, 2008, vol. 46, pp. 99–127.

[19] M. Bosse and J. Roberts, "Histogram matching and global initialization for laser-only SLAM in large unstructured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 4820–4826.

[20] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the 9th International Conference on Computer Vision*, Nice, France, 2003, pp. 1470–1478.

[21] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, October 2008.

[22] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM–FAB-MAP 2.0," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[24] O. Booij, Z. Zivkovic, and B. Kröse, "Efficient data association for view based SLAM using connected dominating sets," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1225–1234, 2009.

[25] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, pp. 3507–3512.

[26] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *The International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–53, August 2010.

[27] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1156–1163.

[28] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the International Conference on Computer Vision Theory and Application*, 2009, pp. 331–340.

[29] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 1828–1833.