2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

IBuILD: Incremental Bag of Binary Words for Appearance Based Loop Closure Detection

Sheraz Khan and Dirk Wollherr
Chair of Automatic Control Engineering
Technische Universität München (TUM), 80333 München, Germany
{sheraz.khan,dw}@tum.de

***Abstract*— In robotics applications such as SLAM (Simultaneous Localization and Mapping), loop closure detection is an integral component required to build a consistent topological or metric map. This paper presents an appearance based loop closure detection mechanism titled 'IBuILD' (Incremental bag of BInary words for Appearance based Loop closure Detection). The presented approach focuses on an *online, incremental* formulation of *binary vocabulary* generation for loop closure detection. The proposed approach does not require a prior vocabulary learning phase and relies purely on the appearance of the scene for loop closure detection without the need of odometry or GPS estimates. The vocabulary generation process is based on feature tracking between consecutive images to incorporate pose invariance. In addition, this process is coupled with a simple likelihood function to generate the most suitable loop closure candidate and a temporal consistency constraint to filter out inconsistent loop closures. Evaluation on different publicly available outdoor urban and indoor datasets shows that the presented approach is capable of generating higher recall at 100% precision in comparison to the state of the art.**

I. Introduction

In the domain of autonomous robotics, loop closure detection within SLAM [1]–[5] is required to build a consistent topological or metric map. In literature, graph based SLAM [1]–[3] consists of two components, the front and the back-end. The front-end deals with the raw laser data and generates node and edge constraints. The back-end estimates the posterior distribution over the robot poses and landmarks given all edge constraints. The loop closure detection mechanism is a component of the front-end of the graph SLAM approach required to generate edge constraints between nodes once the robot returns to a previously visited location. An effective performance of the loop closure detection mechanism is important for SLAM as a single wrong edge constraint can produce an inconsistent map. The importance of an accurate loop closure detection mechanism is further enhanced by the fact that most SLAM back-ends do not filter the generated edge contraints for consistency and leave this up to the front-end. To develop truly autonomous robots that are capable of generating consistent maps, loop closure mechanisms should work at 100% precision while maintaining high recall rate.

In the last few decades, cameras have become an integral part of robotic systems due to an increase in computational power. Inline with this transition the focus of researchers in the robotics community has shifted towards development of appearance based loop closure detection mechanisms [6]–[9]. The advantage of an appearance based approach is that it is capable of detecting loop closures even when the robot pose estimates from odometry might be completely wrong.

sed approach incrementally builds and updates the vocabulary which is based on a binary bag of visual words. In this paper a simple approach for appearance based loop closure detection is presented which builds a vocabulary consisting of *binary visual words* in an *online, incremental* manner by tracking features between consecutive frames. A likelihood function based on inverse occurence frequency of features is used to generate the most suitable loop closure candidates. Additionally, a temporal consistency test is performed to eliminate any incoherent loop closure candidates. The proposed approach is evaluated on different publicly available outdoor and indoor datasets to show its robustness. In addition, a comparison to the state of the art shows that it is capable of producing high recall at 100% precision.

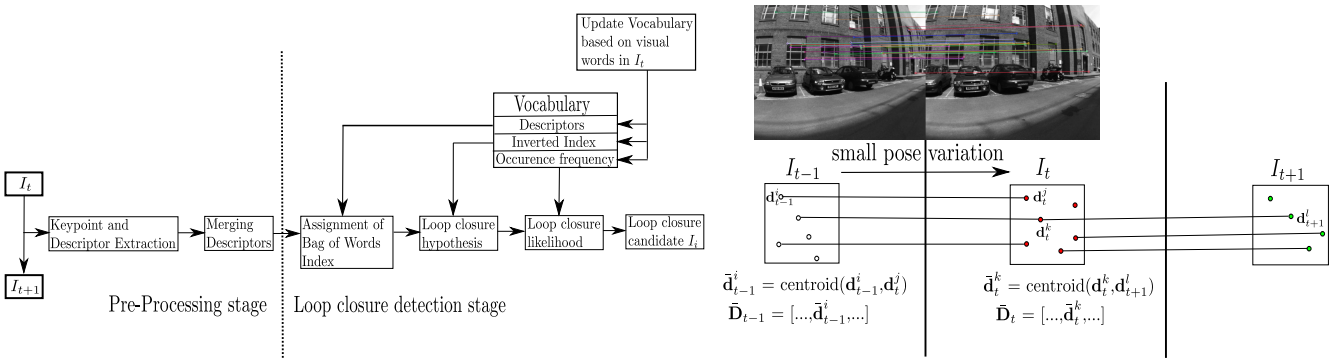
II. Related Work

A. Related work

Majority of the recent work in appearance based loop closure has focused on a bag of words representation [10], [11] for image comparison. Bag of words is a structure that has been borrowed from the field of language processing and information retrieval which allows the representation of an image as a vector by defining the presence or absence of a visual word. The visual words are obtained by clustering descriptors obtained from images after the features extraction process. The literature on appearance based place recognition can be divided into two categories based on the vocabulary generation process: i) Offline and ii) Online, incremental approaches. The work presented in [6], [12], [13] can be placed in the first category whereas [7]–[9], [14] in the second category.

In [6], the authors' propose a probabilistic framework which incorporates co-occurence probabilities between visual words using a Chow Liu tree [15] in an offline vocabulary generation process to perform appearance based loop closure detection. The approach can be considered as the de-facto standard for loop closure detection due to its robustness. In [16], the authors' present an approach that performs loop closure detection and visual odometry using a vocabulary tree generated offline to produce real time

978-1-4799-6923-4/15/$31.00 ©2015 IEEE



(a) Loop closure pipeline

(b) Feature extraction

Fig. 1. (a) A overview of different components of the loop closure detection mechanism. (b) Keypoint extraction and matching mechanism between consecutive images to obtain view point invariant features. $\mathbf{d}^i_t$ represents the i^{th} descriptor extracted at the i^{th} keypoint at time index t.

visual maps. Recently an approach for place recognition using binary bag of words has been presented in [13]. It uses an offline vocabulary learning phase to generate a vocabulary tree consisting of binary visual words. Furthermore, it uses temporal and geometric consistency tests to filter loop closure candidates. In [7], a probabilistic frame work is presented that combines a vocabulary of SIFT [17] features and color histograms for loop closure detection. The features are extracted from a single image and a geometric consistency test based on epipolar constraints is performed to validate loop closure hypotheses. In [8], an approach is presented that incrementally generates a vocabulary using SIFT features matched over a sliding window of images.

The majority of the related work mentioned in the previous section relies on vocabularies which are generated offline, hence are not suitable for robotic applications which require online, incremental operation without any prior training data. Although generic online vocabulary generation [18] mechanisms such as incremental K-means exist, however, they are not well suited for binary spaces because they assume real valued descriptors which can be averaged and rely on the Euclidean distance metric [19]. In contrast, this paper presents a simple approach of *online, incremental binary vocabulary* generation for loop closure detection. The incremental binary vocabulary generation process is based on feature tracking between consecutive frames thereby making it robot pose invariant and ideal for detecting loop closures in real world scenarios. The advantage of using binary descriptors to generate a binary vocabulary is that they offer similar performance to SIFT and SURF features at reduced storage and computational costs [20]. Evaluation of the proposed incremental vocabulary generation process coupled with a simple likelihood function and a temporal consistency constraint shows that it is capable of generating higher precision and recall in comparison to the state of the art on publicly available indoor and outdoor datasets.

III. Pre-processing

The pipeline of operations performed in the loop closure mechanism discussed in this paper is shown in Figure 1(a). The pipeline is composed of two parts: i) Pre-processing ii) Loop closure detection stage. The features extraction (keypoint, descriptor extraction and matching) along with merging of descriptors is placed in the pre-processing step, whereas the rest of the components of the pipeline belong to the loop closure detection stage. This section focuses on the pre-processing components of the pipeline.

A. Feature Extraction

The first step in the pipeline is the extraction of view point invariant features. The approach proposed in this paper uses BRISK (Binary Robust Invariant Scalable Keypoint) features, because they are scale and rotation invariant and offer similar performance to SIFT and SURF at reduced storage and computational costs [20].

The majority of the approaches [6], [7], [13] in appearance based loop closure rely on features extracted from a single image. In contrast, the approach proposed in this paper relies on matching features across consecutive images as shown in Figure 1(b) in a similar manner to [8], [9]. The purpose of matching descriptors across consecutive images (during which the robot undergoes slight variation in its pose) is to determine the most likely descriptors that will be observed in case the robot returns to the same location with a different pose. To match binary descriptors a metric has to be defined to measure similarity. In the proposed approach the Hamming distance is used which is defined as

$$H(\mathbf{d}_t, \mathbf{d}_{t+1}) = \sum_{i=1}^{p} (d_t[i] \oplus d_{t+1}[i]),$$

where $\oplus$ represents the 'exclusive OR' operator and p is the dimension of the descriptor vectors. The index i represents the i^{th} dimension of the p dimensional descriptor vector. $H(*,*)$ represents the Hamming distance whereas $\mathbf{d}_t$, $\mathbf{d}_{t+1}$ are the p dimensional descriptor vectors extracted from image I_t, I_{t+1} respectively *at any keypoint* with t representing the time index. In effect, the descriptors matching process is an 'exclusive OR' between the bits of the descriptor vectors and a count of set bits along the entire descriptor dimension. Two descriptors matched across subsequent images are considered a good match if the Hamming distance between them is below the *matching threshold* δ whereas all

descriptors which do not satisfy this threshold are discarded. The centroid of the matched descriptors is taken as their representative. The centroid $\bar{d}[i]$ of the i^{th} dimension of the binary descriptor vector at *any time index* is calculated as below

$$\forall i \leq p, \bar{d}[i] = \text{centroid}(d^1[i], d^2[i], ..., d^k[i])$$

$$= \begin{cases} 0 & \text{if} \sum_{j=1}^{k}(d^j[i]) < \dfrac{k}{2} \\ \\ 1 & \text{if} \sum_{j=1}^{k}(d^j[i]) \geq \dfrac{k}{2} \end{cases}, \qquad (1)$$

where the notation $d^j[i]$ represents the i^{th} dimension of the j^{th} descriptor vector and k represents the total number of descriptors whose centroid is being calculated. Expression (1) calculates the centroid for any arbitrary number of inputs k, however, in the proposed approach the centroid is calculated for descriptors matched during consecutive time indices as shown in Figure 1(b) and stored in $\bar{\mathbf{D}}_t$ at time index t.

B. Merging Descriptors

The second step in the pre-processing stage as shown in Figure 1(a) is merging descriptors extracted in the previous step. The objective is to remove multiple instances of similar descriptors in case the image contains repetitive patterns. Let $\bar{\mathbf{D}}_t = [\bar{\mathbf{d}}_t^1, \bar{\mathbf{d}}_t^2, ..., \bar{\mathbf{d}}_t^m]^T$ (T and m denote the transpose and the total number of descriptors respectively) represent the centroid of descriptors matched between consecutive images I_t and I_{t+1}. A descriptor after the merging process is termed as a visual word. The algorithm starts by matching a descriptor with all other descriptors in the set $\bar{\mathbf{D}}_t$. Descriptors are merged and replaced by their respective centroid in a *greedy* manner if the distance between them is below the matching threshold δ. This process continues until no further merging can take place. Initially all descriptors in $\bar{\mathbf{D}}_t$ are considered to represent independent visual words, however, after successful merging of descriptors the number of visual words present in the image are reduced. The psuedocode of the merging algorithm is shown in Figure 2. The visual words obtained after merging denoted $\hat{\mathbf{D}}_t$ are then passed to the loop closure detection components of the pipeline.

Merge($\bar{\mathbf{D}}_t$)
Input: $\bar{\mathbf{D}}_t = [\bar{\mathbf{d}}_t^1, \bar{\mathbf{d}}_t^2, ..., \bar{\mathbf{d}}_t^m]^T$
// Descriptors from feature extraction
Output: $\hat{\mathbf{D}}_t$ // Visual words

Procedure:
Initialize number of visual words to m;
// $|\bar{\mathbf{D}}_t|$ represents the number of descriptors
for-all ($i \leq |\bar{\mathbf{D}}_t|$)
  for-all ($j = i+1$ till $j \leq |\bar{\mathbf{D}}_t|$)
    if ($H(\bar{\mathbf{d}}_t^i, \bar{\mathbf{d}}_t^j) < \delta$)
      $\hat{\mathbf{d}}_t^i$ = centroid($\bar{\mathbf{d}}_t^i, \bar{\mathbf{d}}_t^j$);
      $\bar{\mathbf{d}}_t^i = \hat{\mathbf{d}}_t^i$; //update descriptor for next iteration of j
      $\bar{\mathbf{D}}_t - \bar{\mathbf{d}}_t^j$; // remove $\bar{\mathbf{d}}_t^j$ from $\bar{\mathbf{D}}_t$
      decrement m;
      $\hat{\mathbf{D}}_t \longleftarrow \hat{\mathbf{d}}_t^i$; //copy/overwrite i^{th} index in $\hat{\mathbf{D}}_t$
    endif;
  if merging not possible for iteration i
    $\hat{\mathbf{D}}_t \longleftarrow \bar{\mathbf{d}}_t^i$;

Fig. 2. The pseudocode of merging descriptors to remove multiple instances of binary descriptors in the same image

IV. Loop closure detection

This section focuses on the loop closure detection components of the pipeline. The most important component of the loop closure mechanism is the vocabulary. Besides storage of binary visual words in $\mathbf{V}_{t-1}$, the vocabulary consists of two important components:

- Occurence frequency of all binary visual words
- Inverted index for generating loop closure hypothesis

The occurence frequency denoted $\mathbf{F}_{t-1} = [f_{t-1}^1, f_{t-1}^2, ..., f_{t-1}^n]$ contains the number of times a specific visual word is observed in images till time $t-1$. The term n represents the total number of visual words present in the vocabulary. The time index $t-1$ is used here because the update based on the visual words detected in I_t occurs at the end of pipeline, hence after the loop closure likelihood calculations. The vocabulary also maintains an inverted index to generate loop closure hypotheses based on visual words detected in I_t. In the proposed approach the inverted index is stored as a sparse binary matrix which describes the presence or absence of a visual word in all images till time index $t-1$ as shown in Figure 3.

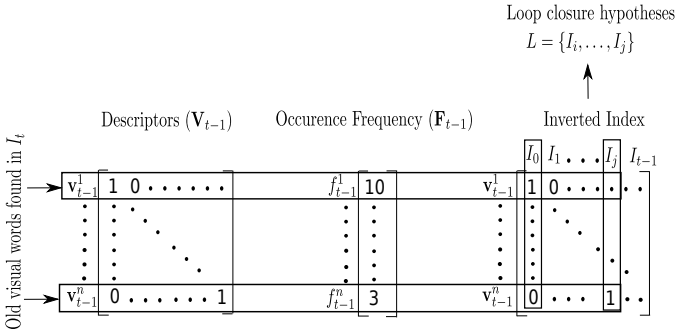


Fig. 3. An overview of the three main components of the vocabulary. The vocabulary consists of the binary visual words stored (row wise) in $\mathbf{V}_{t-1}$ ($n \times p$ matrix), the occurence frequency of each visual word and an inverted index. The inverted index is stored as a sparse binary matrix representing the absence or presence of visual words in all images till time index $t-1$. Given the indices of the old visual words stored in S generated during the assignment of bag of word index, it is possible to determine past images containing the same visual words as shown above.

An overview of the loop closure detection components is mentioned here with further details in the following subsections. After the pre-processing steps, the visual words stored in $\hat{\mathbf{D}}_t$ are assigned a bag of words (*BOW*) index to

Assign-BOW-Index($\hat{\mathbf{D}}_t$, $\mathbf{V}_{t-1}$)
Input: $\hat{\mathbf{D}}_t$ // visual words
$\mathbf{V}_{t-1}$ // visual words in vocabulary at time index $t-1$
Output: S //set of indices of old visual words found in $\hat{\mathbf{D}}_t$
N_{new} // number of new visual words

Procedure:
for-all ($i \leq |\hat{\mathbf{D}}_t|$)
word_found = false;
for-all ($j \leq |\mathbf{V}_{t-1}|$)
if ($H(\hat{\mathbf{d}}_t^i, \mathbf{v}_{t-1}^j) < \delta$)
$S \longleftarrow j$; // store visual word index
word_found = true;
break;
endif;
if ($\sim$ word_found)
increment N_{new};
endif;

Fig. 4. The pseudocode of assigning merged descriptors a BOW index

generate the loop closure candidates. The next step is the evaluation of the likelihood of loop closure candidates and the selection of the best candidate as shown in Figure 1(a). Finally the chosen loop closure candidate is passed through a temporal consistency check. The details of the loop closure detection components are described below.

A. Assignment of the BOW index

The visual words $\hat{\mathbf{D}}_t$ obtained from the pre-processing component of the pipeline are compared with the visual words present in the vocabulary denoted $\mathbf{V}_{t-1}$. This operation is performed to determine the number of the *new* and *old* visual words in $\hat{\mathbf{D}}_t$. The matching threshold δ is used to match the descriptors in $\hat{\mathbf{D}}_t$ with the visual words in the vocabulary to determine the indices of the *old* visual words. The indices of all the *old* visual words are stored in the set S. The pseudocode of the above mentioned process is shown in Figure 4. An important point to mention here is that the vocabulary index $t-1$ is used here because the update based on the visual words detected in I_t occurs at the end of pipeline, hence after the loop closure likelihood calculations. Initially, at time $t=0$ the vocabulary is empty, hence all visual words are initialized as *new* and stored in V_{-1}.

B. Loop closure hypotheses and Likelihood evaluation

Given the set S generated during the assignment of BOW index to the merged descriptors, the loop closure hypothesis set can be generated by using the inverted index. As shown in Figure 3, given the indices of the old visual words detected in I_t and stored in the set S, their occurence frequency and presence in previous images can be easily extracted. A *temporal constraint threshold* β (where $\beta > 0$) is used to prevent the algorithm from generating loop closure hypotheses with images observed close to the current time index. Hence, loop closure hypotheses are limited from time index $t_i = 0$ till t_L. $t_i = 0$ represents the initial time index when the loop closure algorithm started and $t_L = t - \beta$, where t represents the current time index. Let $L = \{I_i, ..., I_j\}$ (where $i \geq 0$ and $j \leq t_L$) represent the set of loop closure hypothesis generated from the inverted index and U represents the set of common visual words between loop closure hypothesis image I_i and currently observed image I_t. The loop closure likelihood of hypothesis I_i with the current image I_t is calculated as

$$\mathcal{L}(I_i, I_t) = \frac{\sum_{\forall m \leq |U|} (f_{t-1}^m)^{-1} |U|}{\sum_{\forall m \leq |U|} (f_{t-1}^m)^{-1} |U| + \sum_{\forall k \leq |T|} (f_{t-1}^k)^{-1} |T| + N_{new}},$$

where T consists of indices of visual words (extracted from the inverted index) present in I_i but not found in I_t. The notation $|T|$ and $|U|$ represents the cardinality of the set. f_{t-1}^j represents the occurence frequency of the j^{th} visual word in the vocabulary. N_{new} is the number of new words detected in I_t. The normalized likelihood of the loop closure candidates is calculated as

$$\hat{\mathcal{L}}(I_i, I_t) = \frac{\mathcal{L}(I_i, I_t)}{\sum_{\forall I \in L} \mathcal{L}(I, I_t)},$$

where L as defined earlier is the entire hypotheses set. The final loop closure candidate is chosen as the maximum of this normalized likelihood function as

$$\max_{\forall I \in L} \hat{\mathcal{L}}(I, I_t).$$

C. Temporal Consistency

The loop closure candidate chosen in the last step of the pipeline goes through a simple temporal consistency test. The temporal consistency test is based on the time index of the previously observed loop closure. Consider a scenario in which a robot at time index $t-1$ returns to a location which was previously visited at time index $t-k$ where $k > \beta$. The temporal consistency test states that after the loop closure event between I_{t-1} and I_{t-k}, all future loop closure events detected in the interval of t and $t+\beta$ are constrained to lie between $t-k$ and $t-k+\beta$. In Figure 5, it can be seen that due to the temporal consistency constraint given the loop closure event at time index $t-1$, the loop closure event between I_t and I_j is rejected (shown in red) whereas the loop closure event in the interval of $t-k$ till $t-k+\beta$ (shown in green) is accepted. In case the robot return to the same location multiple times in the past, the temporal consistency test has to be extended to all such time intervals.

D. Vocabulary Update

Once the likelihood evaluation has taken place, the vocabulary is updated by expanding the vocabulary size based on the number of *new* visual words detected in I_t. Additionally, the occurence frequency of all the old visual words has to be updated and for all the new visual words it has to be initialized to 1. Finally, the inverted index is updated based on the visual words detected in the current time index.

After the vocabulary update the loop closure mechanism waits for the next input image and then repeats the steps of the pre-processing and the loop closure detection stage.

TABLE I

DETAILS ABOUT DATASETS USED IN EVALUATION

Dataset	Description	Camera position	Image size	# Images
Malaga6L [21]	Outdoor, slightly dynamic	Frontal	1024 x 768	3474
City Centre [22]	Outdoor, urban, dynamic	Lateral	640 x480	1237
Lip6 Indoor [7]	Indoor, static	Frontal	240x192	388
Lip6 Outdoor [7]	Outdoor, slightly dynamic	Frontal	240x192	531

TABLE II

RESULTS OF MAGALA6L AND CITY CENTRE DATASET

Dataset	Approach	Precision (%)	Recall (%)
Malag6L	Gálvez-López [13]	100	74.75
	FAB-MAP 2.0 [6]	100	68.52
	IBuILD	**100**	**78.13**
City Centre	Gálvez-López [13]	100	30.61
	FABMAP 2.0 [6]	100	38.77
	IBuILD	**100**	**38.92**

TABLE III

RESULTS OF LIP6 INDOOR DATASET

Dataset	Approach	True positives	False Positives	Ground truth loop closure events
Lip6 Indoor	Angeli [7] (SIFT)	68	0	217
	Angeli [7] (SIFT + Color Histograms)	80	1	217
	IBuILD	**91**	**0**	**217**
Lip6 Outdoor	Angeli [7] (SIFT)	70	0	301
	Angeli [7] (SIFT + Color Histograms)	71	0	301
	IBuILD	**77**	**0**	**301**

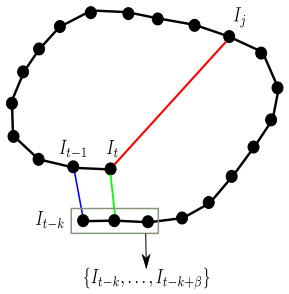


Fig. 5. (Best visualized in color) Given the accepted loop closure at time index $t-1$ (shown in blue), the loop closure at time index t is constrained to lie in $t-k$ till $t-k+\beta$ (shown in green). The loop closure with image I_j (shown in red) is rejected as it does not satisfy the temporal constraint.

V. EXPERIMENTAL EVALUATION

In this section the proposed approach is evaluated on different publically available datasets, as shown in Table I and compared to state of the art methods such as FAB-MAP 2.0 [6], Gálvez-López [13] and Angeli [7] . For all dataset evaluations mentioned in this section the descriptor dimension p is 512 and the temporal constraint threshold β is set to 10. All experiments were performed on an Intel i5-2500K 3.3 GHz processor with 16 GB RAM.

A. Methodology

The correctness of the results for Malaga6L and City centre datasets is established by using the ground truth information and script used by the authors in [13] as a *black box*, hence without any modification in the parameters. The script determines the precision and recall of the algorithm given the ground truth information. The precision of an algorithm is defined as the ratio of correct loop closures to the total number of detected loop closures. The recall is the ratio of the number of correct detections to the ground truth loop closure events. The ground truth information (used in [13]) contains a manually created list of loop closures. *'The list is composed of time intervals, where each entry in the list encodes a query time interval associated with a matching interval'*.

The proposed approach is also compared with Angeli [7] on the Lip6 indoor and outdoor dataset. The ground truth image to image correspondence matrix provided along with the dataset is used to evaluate the number of true and false positives generated by the algorithm. The rows and columns of this matrix correspond to the images at different time indices and an element is set to 1 if loop closure occured and 0 otherwise.

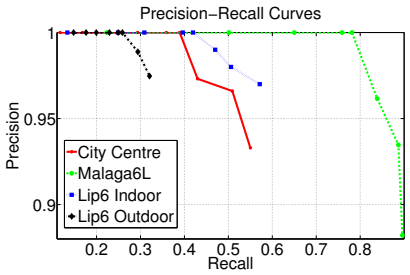
B. Results for City Centre and Malaga6L Dataset

Figure 6(a) shows the precision and recall of the proposed approach for different δ thresholds on the above mentioned datasets. The maximum possible recall rate with 100% precision is mentioned in Table II. The results mentioned in Table II (for FABMAP 2.0 and Gálvez-López) have been taken from [13] as the same script and groundtruth has been used for evaluation. It can be seen that the proposed approach is capable of producing higher recall with 100% precision in comparison to other methods. Figure 6(b) shows the evolution of the vocabulary size for the precision and recall highlighted in Table II. Figure 7 shows the loop closures detected by the approach in red on the City centre and Malaga6L trajectory. Since Malaga6L is the largest dataset (containing 3474 images) used in this paper, the execution

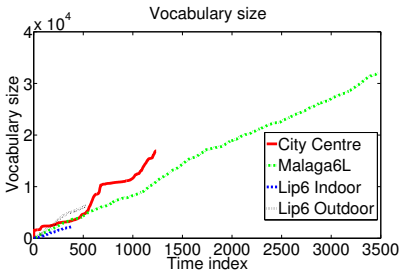
TABLE IV

AVERAGE EXECUTION TIME (MILLI SEC) OF PIPELINE FOR A SINGLE IMAGE ON MALAGA6L DATASET CONTAINING 3474 IMAGES

Property	Keypoint detection	Descriptor extraction	Clustering	Assignment to BOW	Loop closure hypothesis + Evaluation	Vocabulary update
Mean	3.4	1.9	0.038	45	0.1120	0.0088
Standard deviation	0.45	0.44	0.068	37	1.5	0.063

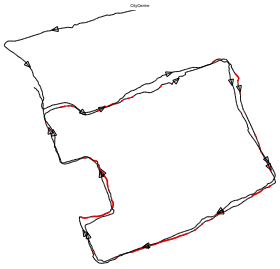


(a) Precision Recall Curves

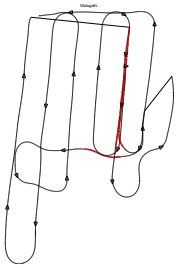


(b) Vocabulary size

Fig. 6. (a) Precision recall curves of the proposed approach (b) Vocabulary size as a function of time for different datasets



(a) Detected loop closures on City Centre dataset



(b) Detected loop closures on Malaga6L dataset

Fig. 7. Loop closures detected (marked in red) by the proposed approach on the map of City Centre and Malaga6L dataset

time of the entire pipeline is mentioned in milliseconds in Table IV. The computation time of the entire pipeline is around 50 millisecond on average per image. Figure 8 shows an example of a loop closure detected by the proposed approach on the City Centre and Malaga6L dataset.

C. Results for Lip6 Indoor and Outdoor Dataset

Figure 6(a) also shows the precision-recall of the proposed approach on the Lip6 Indoor and Outdoor dataset for different δ thresholds. Table III shows the results obtained on the datasets in the same pattern as presented in [7] i.e. greatest number of true positives detected without any false positive whereas Figure 6(b) shows the evolution of the vocabulary size for the highlighted cases in the above mentioned table. It can be seen that the proposed approach is capable of detecting a greater number of loop closures without any false positives. Figure 9 shows an example of the loop closure detected by the proposed approach on the Lip6 Indoor and Outdoor dataset.

VI. DISCUSSION

The experimental evaluation in the previous section raises two important issues about the proposed approach: Firstly, the issue of scalability (handling large vocabularies) and secondly the selection of an appropriate δ threshold.

The first issue is related to the scalability of the proposed approach in context of large vocabularies. In principle this issue can be addressed by formulating an incremental version of the 'vocabulary tree' [23] which is suitable for binary descriptors. The advantage of such an incremental version would be that it would reduce compuational complexity during the BOW assignment process (Section IV-A) and allow the approach to scale well for large scale datasets and vocabularies containing 1 million or more words. A complete discussion and evaluation of such an approach is beyond the scope of this paper and is left as future work.

Consider the second issue of selecting an appropriate δ threshold. The factors that influence the δ threshold include the operating conditions i.e. lighting conditions as current state of the art feature detectors are not completely invariant to such changes and the amount of overlap present between images for feature tracking. In principle, a simple mechanism can be used to estimate the δ threshold for a particular dataset. This mechanism requires matching descriptors (using a specific δ threshold) between a pair of consecutive images and reducing the δ threshold until the false matches are eliminated. It is important that this pair should be a true representative of the operating conditions and expected overlap between images in that dataset.

Fig. 8. An example of loop closure detected by the proposed approach on the City Centre and Malaga6L dataset.

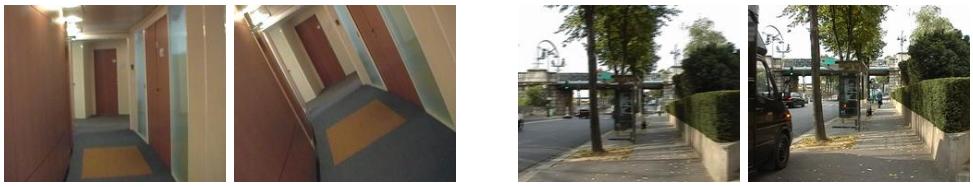


Fig. 9. Different examples of loop closure detected by the proposed approach on the Lip6 Indoor and Outdoor dataset.

VII. Conclusion

In this paper an online, incremental approach of binary visual vocabulary generation for loop closure detection is presented. The proposed binary vocabulary generation process is based on tracking features across consecutive frames making it invariant to the robot pose and ideal for detecting loop closures. An approach for generating and updating the binary vocabulary is presented which is coupled with a simplistic likelihood function to generate loop closure candidates. The proposed approach is evaluated on different publicly available outdoor and indoor datasets. In comparison to the state of the art the proposed approach is capable of generating higher recall at 100% precision.

Acknowledgment

Special thanks to Dorian Gálvez-López for providing details about the results obtained in their paper on the Malaga6L and City center dataset.

References

[1] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[2] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[3] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE, Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *Robotics and Automation Magazine*, vol. 13, no. 99, p. 80, 2006.

[5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *International Joint Conference on Artificial Intelligence*, vol. 18, 2003, pp. 1151–1156.

[6] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[7] A. Angeli, D. Filliat, S. Doncieux, and J. 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, 2008.

[8] A. Kawewong, N. Tongprasit, S. Tangruamsub, and O. Hasegawa, "Online and incremental appearance-based slam in highly dynamic environments," *The International Journal of Robotics Research*, 2010.

[9] S. Khan, D. Wollherr, and M. Buss, "Pirf 3d: Online spatial and appearance based loop closure," in *12th IEEE, International Conference on Control Automation Robotics & Vision (ICARCV)*, 2012, pp. 335–340.

[10] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, 2004, p. 22.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[12] R. Paul and P. Newman, "Fab-map 3d: Topological mapping with spatial and visual appearance," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 2649–2656.

[13] D. Galvez-Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct 2012.

[14] T. Nicosevici and R. Garcia, "Automatic visual bag-of-words for online robot navigation and mapping," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 886–898, Aug 2012.

[15] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[16] 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, 2010.

[17] D. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[18] Y. Girdhar and G. Dudek, "Online visual vocabularies," in *Canadian Conference on Computer and Robot Vision (CRV),*, 2011, pp. 191–196.

[19] M. Muja and D. G. Lowe, "Fast matching of binary features," in *2012 IEEE Ninth Conference on Computer and Robot Vision (CRV),*, 2012, pp. 404–410.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[21] J.-L. Blanco, F.-A. Moreno, and J. González, "A collection of outdoor robotic datasets with centimeter-accuracy ground truth," *Autonomous Robots*, vol. 27, no. 4, pp. 327–351, November 2009.

[22] 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. 647–665, 2008.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*, vol. 2, 2006, pp. 2161–2168.