2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

A Flexible and Efficient Loop Closure Detection Based on Motion Knowledge

Bingxi Liu[1,2], Fulin Tang[2], Yujie Fu[1,2], Yanqun Yang[3] and Yihong Wu[1,2,*]

***Abstract*— Loop closure detection (LCD) is an essential module for simultaneous localization and mapping (SLAM), which can correct accumulated errors after long-term explorations. The widely used bag-of-words (BoW) model can not satisfy well the requirements of both low time consumption and high accuracy for a mobile platform. In this paper, we propose a novel LCD algorithm based on motion knowledge. We give a flexible and efficient detection strategy and also give flexible and efficient combinations of a global binary feature extracted by convolutional neural network (CNN) and a hand-crafted local binary feature. We take a continuous motion model, grid-based motion statistics (GMS) and motion states as motion knowledge. Furthermore, we fuse the proposed LCD with a visual-inertial odometry (VIO) system to correct localization errors by a pose graph optimization. Comparative experiments with state-of-the-art LCD algorithms on typical datasets have been carried out, and the results demonstrate that our proposed method achieves quite high recall rates and quite high speed at 100% precision. Moreover, experimental results from VIO further validate the effectiveness of the proposed method.**

I. INTRODUCTION

SLAM has received much attention because of its wide applications in robot navigation and augmented reality (AR) [1]. Although plenty of SLAM algorithms have been proposed and have surprising performance, after long-time explorations, their estimated trajectories and maps inevitably contain accumulated errors [2], [3]. Visual loop closure detection (LCD) is a recognized solution to this problem, which can be considered as an on-line image retrieval problem that matches the scene in the current location with previously visited locations [4].

Feature extraction and clustering are the core of visual LCD algorithm. Hand-crafted features can be divided into global and local features. Global features [5], [6] are compact descriptions of image contents, which are sensitive to view changes. By contrast, local features [7] can deal with view changes and occlusions, but the robustness may be worse. As a local feature clustering technology, bag-of-words (BoW) model [8] improves the computational efficiency, and it is widely used in LCD [2], [9]–[15]. The vocabulary obtained through unsupervised training is used to quantify features into visual words. Then, the technique of term frequency-inverse document frequency is widely used to create corresponding vectors depicting the histogram of visual words [8]. However, because the histogram description is lack of the spatial distribution of visual words, the BoW model often recalls false positive results. To alleviate this problem, some researchers introduce geometrical verification and temporal consistency constraint to LCD [9], [10].

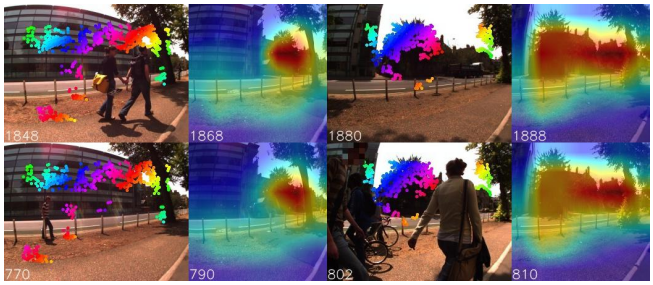

Fig. 1. The query images (Top) and the loop closure detection images (Bottom) by the proposed method. The local features of matched GMS are visualized as color points and the output features of trained CNN are visualized as heat maps by Grad-CAM [16]. The time stamp difference conforms to the principle of temporary consistency.

In addition to hand-crafted features, features extracted by convolutional neural networks (CNN) are also divided into global features [17], [18] and local features [19], [20]. In visual place recognition, the comprehensive performance of the features extracted by CNN surpasses the features extracted by traditional methods. CNN features improve abilities to deal with repeated textures [17] and illuminations [19], which are difficult to be solved by traditional methods.

The BoW model still can not achieve a high recall rate and high computation speed simultaneously despite of continuously improvement. Relying on the supervision information of GPS labels or manual labels, CNN-based LCD systems become troublesome in actual use. More importantly, almost all LCD algorithms concern with more individual image representations and less motion knowledge to improve their performances in terms of recall and speed.

To address these problems, we present a novel visual LCD algorithm based on motion knowledge, which is precise and fast. The motion knowledge includes a continuous motion model, grid-based motion statistics (GMS) and motion states. The novel contributions are as follows:

1) We propose a self-supervised CNN learning method suitable for LCD, which extracts global binary features. The features can not only make fast searches but also simultaneously measure well scene similarities.

* This work was supported by National Natural Science Foundation of China under Grant Nos. 61836015, 62002359 and supported by the Beijing Advanced Discipline Fund under Grant No. 115200S001. The corresponding authors are Fulin Tang and Yihong Wu. E-mail: fulin.tang@nlpr.ia.ac.cn and yhwu@nlpr.ia.ac.cn

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.

[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

[3]Innovation Technology Center, Shanxi Coking Coal Group Co., LTD, Taiyuan, China.

2) A novel geometrical verification method based on motion statistics is proposed, where local features are used to handle view changes and occlusions with a fast speed.
3) Based on motion states, a flexible and efficient detection strategy is proposed. And then different combinations of the above global features 1) and local features 2) are provided.
4) Using the LCD in the above 3), a pose graph optimization is designed to correct accumulated errors for a visual-inertial odometry (VIO) system.
5) Experiments show that the proposed method outperforms state-of-the-art LCD methods and achieves quite high recall rates and quite high speed at 100% precision at the same. Fig. 1 shows some results of our LCD system.

The rest of this paper is organized as follows. In Section II, we introduce the related work. We describe the proposed methods in Section III. Section IV presents the experimental results and analysis, followed by conclusions in Section V.

II. RELATED WORK

Image retrieval, place recognition and LCD are three very similar problems but have differences. Image retrieval is targeted at a variety of images, including people, animals, buildings, and so on. It is a very large scale retrieval and runs on a server [21]. Place recognition can be subdivided into two categories, returning similar images from databases [18] and returning place description [22]. The former has more researches for visual localization [23] and is the most similar to LCD. But its difference with LCD is that LCD considers a shorter time duration than place recognition, such that the problem of season long-term changes [24] needn't be considered in LCD. However, the images of LCD are continuous in space-time dimensions. The challenge of LCD is to distinguish similarities under ambiguous recognition and to maintain less storage and faster computing speed for operations on moving platforms usually with low computation sources.

Originating from text retrieval, BoW model based on unsupervised training was firstly introduced into image retrieval area by Zisserman [8]. Later, BoW model becomes the most frequently used core algorithm among visual LCDs. These methods can be distinguished into two categories according to their visual vocabulary construction procedures. The first category are the methods that use pretrained vocabularies [2], [10], [12], and the second category are the methods that build visual vocabularies on-line [9], [11], [13], [15].

In the first category, one of the most influential methods is FAB-MAP 2.0 [10], which uses a Chow Liu tree to learn a generative model of place appearance. Gálvez-López et al. [2] proposed a hierarchical BoW model built from binary features that supports direct and inverse indexing, and uses geometrical verification to eliminate false recalls.

On the other hand, Angeli et al. [9] used an incremental BoW model to estimate the matching probability through Bayesian filtering. In [11], an incremental vocabulary building process is presented using agglomerative clustering. Moreover, Khan et al. [13] proposed an on-line formulation of binary vocabulary, with binary features between consecutive images. Tsintotas et al. [15] proposed a dynamic segmented input image flow and used a voting scheme to determine the appropriate candidate position based on the online generated visual words.

With the development of deep learning, some CNN-based LCD methods are gradually proposed. An et al. [25] proposed an incremental LCD algorithm using a proximity graph structure for searching with real-valued global features extracted by pre-trained CNN, and extracted SURF features for geometrical verification. In [26], a LCD algorithm was proposed, which also uses a graph-based visual place recognition method and the features extracted from CNN.

Supervised training is required for almost all CNN-based LCD methods. Although these pre-training models have some generalization, they are limited to datasets with GPS labels [18] or manual labels [22], [27] compared to BoW model with unsupervised training. We propose a self-supervised learning method to extract global binary features without manually labeling data and use a novel geometrical verification method with feature motion statistics. Another difference from all the above methods is that they use tree or graph structures to storage features, while we use a simplest linear storage that conforms to the logic of motions by which a flexible strategy to combine different features can be designed.

III. PROPOSED METHOD

As shown in Fig. 2, we provide an overview of our proposed framework for LCD, which consists of a self-supervised learning method and an on-line LCD pipeline. Additionally, the pipeline of loop closure correction is also shown in Fig. 2.

The on-line LCD pipeline is divided into two modes: robust and lazy. The robust mode consists of three steps: global binary feature retrieval, geometrical verification based on motion statistics and temporal consistency check. When a loop is detected in the robust mode, lazy mode flexibly uses global binary features or local binary features to verify the hypothetical loops based on temporal consistency. Furthermore, the proposed LCD is used to correct accumulated errors in a long-term exploration for a VIO system.

A. Self-supervised Learning with Continuous Motion Model

We define a continuous motion model as follows. A vision robot moves at a stable speed in a single direction, and obtains a continuous image sequences $X = \{x_i\}_{i=1}^{N}$, as shown in A of Fig. 2. The number of coming images increases with time, but the change direction of similarity degree is opposite. The image x_t is similar to x_{t+1}. Compared with images that are farther away from time, x_t and x_{t+d} are considered dissimilar. Based on the above model, there are N images that can generate M triplet label $T = \{q_i, p_i, n_i\}_{i=1}^{M}$. Here q_i is the query image sample in the i th triplet, p_i is the positive image sample, and n_i is the negative image sample.

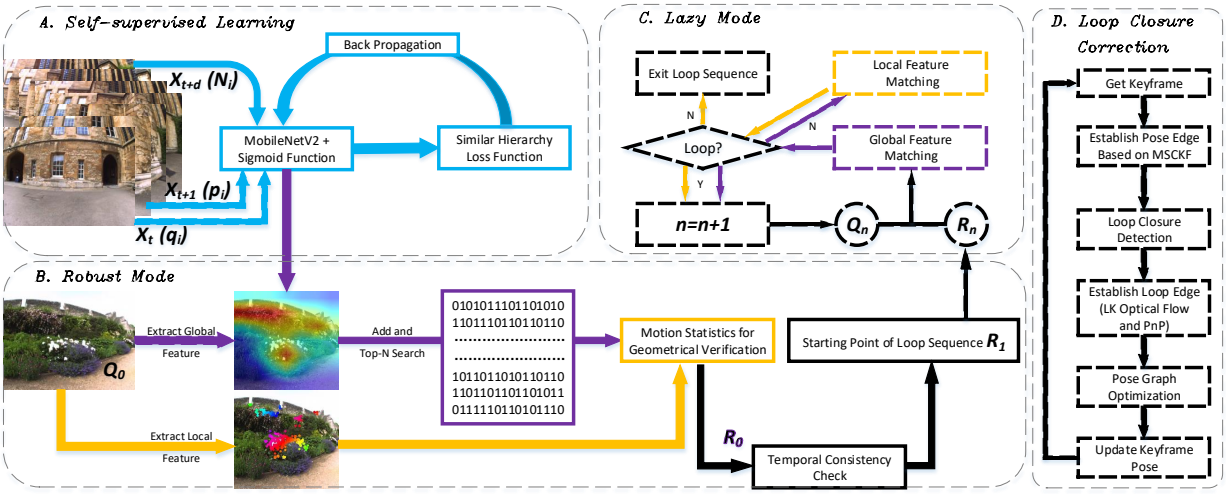

Fig. 2. An overview of our proposed LCD framework and loop closure correction.

Through the timestamp of the continuous motion model, it is not necessary to manually mark the image dataset or the image acquisition equipment equipped with GPS. Therefore, it is very convenient to train a place recognition model suitable for LCD. The training data acquisition needs to follow the rules:

1) The camera moves towards a single direction.
2) The camera makes a uniform or nearly uniform motion.
3) There is no long-term loop.

We design a triplet hashing network, namely three feed-forward feature extraction sub-networks, whose parameters are shared and the last activation function is set to a Sigmoid function. Triplet images T are input to the network, which aims to learn the similarity of q_i and p_i and the dissimilarity of q_i and n_i.

MobileNetV2 [28] is selected as the feature extraction part, because its efficient reasoning capability, based on inverted residual blocks and linear bottlenecks, is very suitable for deployment on mobile terminals. The loss function used here is a hierarchical similar loss function based on a probability for learning highly quality hash codes.

B. Grid-based Motion Statistics for Geometrical Verification

It is not robust to solely rely on global binary features and thus the local feature matching based on grid-based motion statistics (GMS) [29] is used as a geometrical verification. The local feature used in our system is ORB, which consists of the improved FAST keypoint and the oriented BRIEF descriptor.

For a pair of images from different views in the same 3D scene, a feature correspondence means that a feature in one image is identified as the feature in another image projected from the same 3D point. Assuming that the motion process is smooth, neighboring features move together. True correspondences are influenced by the smoothness constraints, while false correspondences are not. Therefore, true correspondences often have more similar neighbors than false correspondences.

As shown in Fig. 3, images I_1 and I_2 are divided into non-overlap grids, respectively. Assume c_i be a correspondence that lands in the grid G_a and G_b. We define c_i's similar neighbors as:

$$S_i = \{c_j | c_j \in C_{ab}, c_i \neq c_j\}, \tag{1}$$

where C_{ab} are those correspondences landing in G_a and G_b simultaneously.

We term $|S_i|$, the number of elements in S_i, motion support for c_i. The motion support can be used as a discriminative feature to distinguish true and false correspondences. This is based on the motion smoothness hypothesis, which is a key reason why (1) is more suitable for LCD in a SLAM system.

C. Flexible Detection based on Motion States

When a robot reaches to a looping point, it will move on a looping path for a period of time. Therefore, we divide a motion state into a non-loop state and a loop state, as shown in Fig. 4. Assuming that the query image Q_0 detects the looping image R_0, other frames Q_i detect R_i, with $i = 1...n$. Because of the discrete storage structure, the previous methods can only use the principle of temporal consistency as a means of checking the quality of loops. Distinguishing these motion states, the proposed LCD, by a linear storage and by combining the global and local features, has two modes as shown in B and C of Fig. 2: a robust mode and a lazy mode.

1) Robust Mode: Q_0 is extracted as the global binary feature by the given in A of this section. The global binary

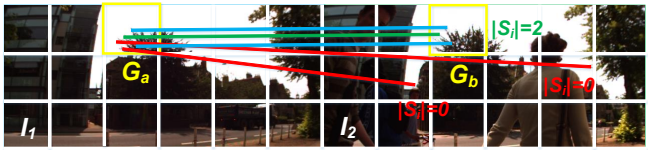

Fig. 3. True correspondences often have more similar neighbors than false correspondences, so the number of similar neighbors is counted.

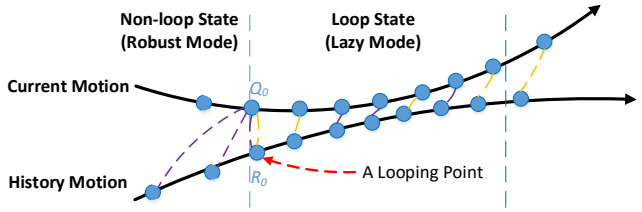

Fig. 4. A schematic diagram of a loop. These purple lines represent detection using global features, while these yellow lines represent local features matching.

feature, which means the hash code, is added to the end of the hash code book, and Top-N brute force search is carried out in a Hamming space. More specifically, the Hamming distance between Q_0 and the previous keyframes is calculated, and the images corresponding to the hash code with the minimum Hamming distance and less than the threshold δ_1 are selected as the candidate results.

After the candidate results are returned in the above process, local features are extracted from the query image and candidate results. And then geometrical verification based on GMS between these pairs is performed by the given method in B of this section. The image with the largest inlier rates greater than a parameter γ_1 is selected as the candidate R_0.

Then we perform a temporal consistency check within 2 frames, that is to perform GMS matching again on the next image Q_1 and its candidate R_1. The assumption of temporal consistency is an ideal. As shown in Fig. 4, the difference of current two continuous frames and the difference of their candidate frames are sometimes inconsistent. Moreover, the motion directions are inconsistent too. This means that the similarity between Q_1 and R_1 is more likely to be lower than that of Q_0 and R_0. And thus a lower bound parameter of the inlier point rate is set here, which is denoted as γ_2. If R_0 and R_1 pass the above process, the system will accept them as the loop result, and enter the lazy mode.

2) Lazy Mode: After a period of time N, image Q_n is preferentially matched with image R_n by the global CNN hash features of Section III-A. If the distance between the pair of hash codes is greater than the parameter δ_2, that is, global matching doesn't work, then GMS matching given in Section III-B is performed, where the taken lower bound parameter is γ_3. All parameter relationships are expressed as follows:

$$\begin{aligned} &0 < \delta_1 < \delta_2 \ll length(HashCode) \\ &0 < \gamma_3 < \gamma_2 < \gamma_1 \ll 1 \end{aligned} \tag{2}$$

If both the global and local matching methods fail, the lazy mode is exited. Very few false positive results appear at the end of loop sequence and then a reverse temporal consistency check is performed when exiting.

D. Pose Graph Optimization

We select a VIO (MSCKF) [30] as the front-end of our SLAM system to get poses of all image frames. Once a loop closure is detected, we carry out a pose graph optimization to correct poses. In order to reduce the computational cost, the system only performs LCD and pose graph optimization on keyframes. The pipeline of the pose graph optimization is shown in D of Fig. 2.

First, according to parallaxes, keyframes are selected in the SLAM system. Then, we construct a pose graph. Each node represents the pose of a keyframe and an edge represents the relative pose between two keyframes. If a loop is detected, we carry out LK optical flow between the loop keyframe and the current keyframe to get 3D-2D correspondences. Therefore, the relative pose between the loop keyframe and the current keyframe is obtained. Finally, a pose graph optimization is performed to correct the accumulated errors.

The objective function of the pose graph optimization is as follows:

$$\min \sum_{(i,j)\in\varepsilon} \left\| \boldsymbol{t}_{ij} - \boldsymbol{R}_i^T(\boldsymbol{t}_j - \boldsymbol{t}_i) \right\|_2^2 + \left\| \boldsymbol{R}_{ij} - \boldsymbol{R}_i^T \boldsymbol{R}_j \right\|_F^2, \tag{3}$$

where $\{\boldsymbol{R}_i\} \in SO(3), \{\boldsymbol{t}_i\} \in R^3$. Here $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ represent the rotation matrix and translation vector of the camera associated with frame i under the world coordinate system. $\boldsymbol{R}_{ij}$ and $\boldsymbol{t}_{ij}$ represent the relative rotation and translation between frames i and j. ε represents the set of edges in the pose graph, and (i,j) represents an edge connecting frame i and frame j.

The rotation matrix $\boldsymbol{R}$ in (3) is dominant [31] and the second error term in the above formula can be considered first. At first, we solve an unconstrained optimization as follows:

$$\min \sum_{(i,j)\in\varepsilon} \left\| \boldsymbol{R}_{ij} - \boldsymbol{R}_i^T \boldsymbol{R}_j \right\|_F^2. \tag{4}$$

This is a linear least square problem, whose solution can be computed efficiently. It is possible that the solution $\boldsymbol{R}_i^*$ is not a rotation matrix, which needs to be corrected. We compute the singular value decomposition of $\boldsymbol{R}_i^*$ and then get:

$$\boldsymbol{R}_i = \boldsymbol{S} \cdot diag\begin{bmatrix} 1 & 1 & det(\boldsymbol{S}\boldsymbol{V}^T) \end{bmatrix} \cdot \boldsymbol{V}^T. \tag{5}$$

Once we obtain $\boldsymbol{R}_i$ according to (5), we take $\boldsymbol{R}_i$ as an initial value to solve (3).

IV. EXPERIMENTS

To evaluate the proposed LCD method, a series of experiments are carried out on publicly available datasets, including New College, City Centre [10] and KITTI 00, KITTI 05 [32]. The ground truths of KITTI datasets are provided by the authors in [15]. The pose graph optimization is designed based on a VIO system, but there is no inertial sensor data in these public datasets. Therefore, we build CASIA View dataset to test the effect of the pose error correction by the proposed LCD. A more detailed description of these datasets can be seen in Table I. We only use a small part of the entire dataset for self-supervised learning, such as 50% of the New College and City Centre datasets, 25% of the KITTI 00 and 05 datasets, and 6% of the CASIA View datasets.

TABLE I

DESCRIPTIONS OF THE USED DATASETS

Dataset	Description	Acquisition Method	Image Characteristics	Sequences Size
City Centre	Outdoor, Dynamic	Mobile Robot	640 × 480, 0.5Hz	2474 images, apprx.2 km
New College	Outdoor, Dynamic	Mobile Robot	640 × 480, 0.5Hz	2146 images, apprx.1.9 km
KITTI 00	Outdoor, Dynamic	Automobile	1241 × 376, 10Hz	4551 images, apprx.11 km
KITTI 05	Outdoor, Dynamic	Automobile	1226 × 370, 10Hz	2761 images, apprx.7.5 km
CASIA View	Outdoor and Indoor, Dynamic	Electric Bicycle and Walk	640 × 480, 25Hz	100373 images, n.a

In this section, we firstly explain some details of our LCD algorithm through the experiments on the used parameters. Then the performance of the pose graph optimization using the detected loops is presented. Following this, the performance of the LCD method is compared against state-of-the-art methods. Finally, the execution time and hardware usage are analyzed.

A. Performances of the used parameters

The global feature dimension is set to 512 and the number of local feature extraction is set to 500. Behaviors of these parameter variations have been analyzed in other papers and thus aren't done here. Here there are six key parameters to be discussed in the following experiments.

Top-N brute force search for the global features involves two parameters δ_1 and N that could affect the search quality, where δ_1 is the upper bound of Hamming distance in the robust mode and N is the maximum number from the result of Top-N brute force search. Similarly, δ_2 is the upper bound of Hamming distance for the global matching. When $\delta_2 = 0$, it is equivalent to having no global feature matching in the lazy mode.

As described in Section III-C, our system has three parameters for the local feature matching, where γ_1 and γ_2 are set in the geometrical verification phase and temporal consistent phase, respectively. The best value relationship among them should be: $\gamma_1 > \gamma_2 > \gamma_3$, and when $\gamma_3 = 1$, the local feature matching is turned off in the lazy mode. The parameter $\gamma_1 = 0.2$ is the key parameter of the system, which can basically guarantee a higher precision. Let's use it first and explain it later.

Firstly, by choosing different δ_1 and δ_2, we perform the experiments on the New College dataset. The other parameters are set as: $\gamma_1 = 0.2, \gamma_2 = 0.14, \gamma_3 = 0.08, N = 3$. The recalls at 100% precision and mean time consumptions are shown in Fig. 5. As δ_1 increases, the recall rate increases significantly but the calculation speed slows down. As δ_2 increases, not only the recall rate increases slightly, but also the calculation speed increases. We choose $\delta_1 = 14$ and $\delta_2 = 24$ in the later experiments.

Secondly, N is evaluated. In Table II, it can be found that as N increases, the recall rate first increases and then decreases. The decline is due to more choices leading to a worse starting point of a loop. In order to make the proposed method faster, we choose $N = 1$.

Finally, we evaluate the lower bounds γ_1, γ_2, γ_3 of the inlier rate for the local feature matching. In the first three rows of Table III, we set $\gamma_2 = 0.10$ and $\gamma_3 = 1$, which mean a weaker

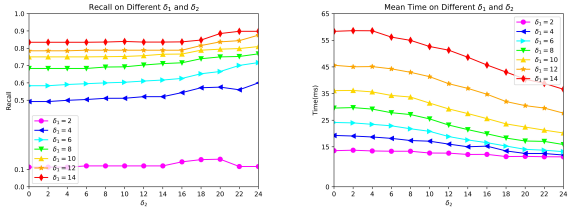

Fig. 5. The recall at 100% precision of our algorithm on the New College dataset using different δ_1 and δ_2. (Left) The average execution time of our algorithm on the New College dataset using different δ_1 and δ_2. (Right)

TABLE II

PERFORMANCES ON NEW COLLEGE DATASET WITH DIFFERENT N

N	1	3	5	7
Recall(%)	89.91	92.14	91.55	91.55
Mean Time(ms)	26.21	30.03	30.88	31.26

temporal consistency check and turning off GMS matching in the lazy mode. As γ_1 increases, the accuracy rate increases, and the recall rate first increases and then decreases. The improvement of the recall rate is because the found loop starting point becomes more accurate, thus increasing the robustness of the lazy mode. The drop is because those true positive results with low similarities are eliminated. As shown in Table III, with the increases of γ_2 and γ_3, the recall rate rises slowly.

TABLE III

PERFORMANCES ON NEW COLLEGE DATASET WITH DIFFERENT $\gamma_1, \gamma_2, \gamma_3$

γ_1	0.08	0.12	0.16	0.20	0.24
Precision(%)	98.97	100	100	100	100
Recall(%)	84.75	88.51	88.51	86.17	85.23

γ_2	0.14	0.12	0.10	0.08	0.06
Precision(%)	100	100	100	100	100
Recall(%)	86.17	86.17	87.69	87.69	87.69

γ_3	1	0.08	0.06	0.04	0.01
Precision(%)	100	100	100	100	100
Recall(%)	87.69	90.97	90.97	91.21	91.21

B. Evaluation on CASIA View dataset

The LCD and pose graph optimization algorithms are tested in the CASIA View dataset. In Table IV, experimental results are quantified by mean optimization time consumptions and reprojection errors. Obviously, the time consumptions satisfy the real-time requirement for robot

TABLE IV

RESULTS OF THE POSE GRAPH OPTIMIZATION ON CASIA VIEW DATASET

Stages	Outdoor1	Outdoor2
Number of Keyframes	1101	432
Total Optimization Time (ms)	98.14	33.24
Mean Optimization Time (ms/keyframe)	0.029	0.077
Reprojection Error (pixel)	1.53	1.83

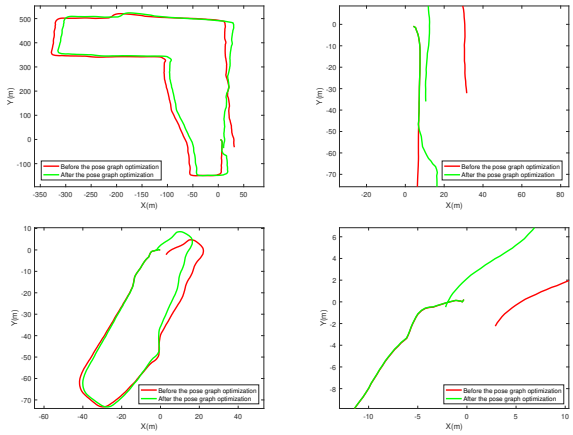

Fig. 6. The proposed loop closure detection and error correction are tested on the CASIA View dataset. The global trajectory (Left) and the local trajectory (Right) at the closed-loop are shown respectively.

localizations. And the reprojection errors are smaller than 3.0 pixel, which proves that the pose graph optimization is accurate and the proposed LCD is effective. Intuitively, Fig. 6 shows trajectories before and after the pose graph optimization. After the pose graph optimization, accumulate errors are corrected.

C. Evaluation on public datasets including comparisons

In Table V and VI, the recall at 100% precision and the mean execution time of the proposed algorithm against the famous or state-of-the-art methods are compared. The best results are set to bold font. We see that the proposed method performs best on New College and City Centre datasets, in particular increasing 14.47% from the second best on New College. The recalls of the proposed method is slightly lower than the best ones, of Tsintotas's method [15], on the KITTI datasets. While the execution time is only one-fifteenth of [15]. More importantly, the proposed method is the most stable one.

D. Execution time, storage usage and memory usage

The proposed method is carried out on a Intel(R) Xeon(R) W-2135 CPU machine with a NVIDIA 2080TI GPU card. The specific step execution time consumed per image is shown in Table VII. The main factor that affects execution time on different hardware is the global feature extraction. On a Huawei P30 mobile phone, we test the feature extraction time of MobileNetV2, which is only about 45ms. In our method, a 560 MB dataset is saved as the hash code book with only 1.4 MB and the size of CNN model is only 12.8 MB. By our method, the memory usage without any optimization is about 2 GB and is only one-fourteenth of An's method [25].

TABLE V

MAXIMUM RECALLS OF DIFFERENT ALGORITHMS AT 100% PRECISION

	City Centre[1]	New College[1]	KITTI 00[2]	KITTI 05[2]
FAB-MAP 2.0 [10]	40.11	52.63	61.22	48.51
SeqSLAM 2.0 [33]	75.12	66.67	78.33	61.48
iBoW-LCD [14]	82.03	53.03	76.50	53.07
Kazmi et al. [34]	75.58	51.09	90.39	81.41
DLoopDetector [2]	30.59	47.56	72.43	51.97
Tsintotas et al. [15]	52.44	16.30[1]	**93.18**	**94.20**
An et al. [25]	66.48	76.74[1]	91.23	85.15
Proposed	**86.01**	**91.21**	93.02	92.53

[1] City Centre and New College datasets have official ground truths and thus all methods can be compared. Note that the number of images in the New College used by [15] and [25] is different from other methods.

[2] KITTI has no official loop ground truths and so many researchers [15], [26], [35] have annotated it by themselves. The used ground truths in [2], [15] and [25] are the same as ours.

TABLE VI

MEAN EXECUTION TIME OF DIFFERENT ALGORITHMS (MS)

	City Centre	KITTI 00
FAB-MAP 2.0 [10]	259.7	388.1
SeqSLAM 2.0 [33]	97.9	107.8
iBoW-LCD [14]	175.5	277.1
Kazmi et al. [34]	95.4	96.9
DLoopDetector [2]	**27.51**	111.04
Tsintotas et al. [15]	183.23	521.54
An et al. (GPU) [25]	40.23	62.68
Proposed (GPU)	33.34	**35.42**

TABLE VII

MEAN EXECUTION TIME OF LCD ON CITY CENTRE DATASET

Stages	GPU (ms/query)	CPU (ms/query)
Global Feature Extraction	4.47	19.11
Top-N Search	2.80	2.90
ORB + GMS Match	21.57	21.51
Whole System	33.34	48.04

V. CONCLUSIONS

In this paper, we propose a novel LCD method based on motion knowledge. Self-supervised learning based on a continuous motion model is proposed and CNN global binary features are used for fast global searches. The candidate loops are verified by local feature motion statistics. The different combinations of the two features, as well as fully using motion states, enable us to construct a flexible and efficient detection strategy. The method is evaluated on publicly available outdoor datasets as well as dataset captured by ourselves, and the results show that it achieves quite high recall rates and quite high speed at 100% precision simultaneously. Moreover, experimental results from a VIO system further validate the effectiveness of the proposed method, which can be applied well on a mobile intelligent agent.

VI. ACKNOWLEDGMENT

The authors wish to gratefully acknowledge Dr. Konstantinos A. Tsintotas for kindly offering ground truth information for dataset, and Dr. Shan An for his kindly help.

REFERENCES

[1] F. Tang, H. Li, and Y. Wu, "Fmd stereo slam: Fusing mvg and direct formulation towards accurate and fast stereo slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 133–139.

[2] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics (TRO)*, vol. 28, no. 5, pp. 1188–1197, 2012.

[3] Y. Wu, F. Tang, and H. Li, "Image-based camera localization: an overview," *Visual Computing for Industry, Biomedicine, and Art*, vol. 1, no. 1, pp. 1–13, 2018.

[4] S. Lowry, N. Snderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics (TRO)*, vol. 32, no. 1, pp. 1–19, 2016.

[5] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics (TRO)*, vol. 25, no. 4, pp. 861–873, 2009.

[6] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 1023–1029.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[8] Sivic and Zisserman, "Video google: a text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1470–1477 vol.2.

[9] 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 (TRO)*, vol. 24, no. 5, pp. 1027–1037, 2008.

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

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

[12] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 846–853.

[13] S. Khan and D. Wollherr, "Ibuild: Incremental bag of binary words for appearance based loop closure detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5441–5447.

[14] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3051–3057, Oct 2018.

[15] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "Assigning visual words to places for loop closure detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1 – 7.

[16] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[17] L. Wu and Y. Wu, "Deep supervised hashing with similar hierarchy for place recognition." in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3781–3786.

[18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

[19] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *IEEE international Conference on Computer Vision (ICCV)*, 2017, pp. 3456–3465.

[20] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 224–236.

[21] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin, "Visual search at alibaba," in *ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, 2018, pp. 993–1001.

[22] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 40, no. 6, pp. 1452–1464, 2017.

[23] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *The British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2012, p. 4.

[24] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *International Journal of Robotics Research (IJRR)*, 2016.

[25] S. An, G. Che, F. Zhou, X. Liu, X. Ma, and Y. Chen, "Fast and incremental loop closure detection using proximity graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 378–385.

[26] X. Zhang, L. Wang, Y. Zhao, and Y. Su, "Graph-based place recognition in image sequences with cnn features," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 389–403, 2019.

[27] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2575–2584.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[29] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4181–4190.

[30] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.

[31] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization," in *IEEE international conference on robotics and automation (ICRA)*, 2015, pp. 4597–4604.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[33] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.

[34] S. M. A. M. Kazmi and B. Mertsching, "Detecting the expectancy of a place using nearby context for appearance-based mapping," *IEEE Transactions on Robotic (TRO)*, vol. PP, pp. 1–15, 07 2019.

[35] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Bronte, "Fast and effective visual place recognition using binary codes and disparity information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3089–3094.