This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

Image Sequence Matching Using both Holistic and Local Features for Loop Closure Detection

Yicheng Li, Zhaozheng Hu, IEEE Member, Gang Huang, Zhixiong Li, IEEE Member, Miguel Angel Sotelo, IEEE Senior Member

***Abstract*—Simultaneous localization and mapping (SLAM) has a wide range of applications, such as mobile robots, intelligent vehicle localization and intelligent transportation system (ITS). However, loop closure detection is a challenge task for SLAM. This task concerns the difficulty of recognizing already mapped areas. To this end, this paper proposes a novel loop closure detection method called image sequence matching (ISM), which only uses a low-cost monocular camera. This method first divides the already mapped areas into some "feature-zones". One feature-zone is selected by a novel topological detection model. Then, we adopt two different feature spaces to make sequence matching between query image and feature-zone. Last but not least, we propose a novel clustering method called voting K-nearest neighbor (V-KNN) to fuse candidates. As a result, the ISM method has been validated by using collection datasets and public datasets, which were collected along different routes, covering different times and weather conditions. The total lengths of these routes are more than 10 km. Experimental results show that the ISM method can adapt to different times with good detection stability in varying scenarios. The mean of detection errors are all less than 1 frame and the detection accuracies are all more than 90% in these scenarios. Compared to other methods, the proposed method has high accuracy and great robustness.**

***Index Terms*— SLAM; loop closure detection; image sequence matching; feature-zone; topological detection; V-KNN**

I. INTRODUCTION

For the past decades, with the rapid development of science and technology, mobile robots are gradually realizing automation. An exact estimation of the robot position is the basis for achieving this goal [1]. In this context, simultaneous localization and mapping (SLAM) is becoming a hot topic all over the world [2][3]. Traditionally, this approach has depended on range and bearing sensors such as laser scanners, radars, etc. Besides, with the rapid development of computing power, cameras with low cost are extensively used in SLAM nowadays. For instance, in previous work, Whelan et al. [4] present a SLAM system with a low-cost RGB-D camera. This system has a capable of producing high-quality consistent surface reconstructions. Similarly, Chen et al. [5] set up multi-robot ceiling vision SLAM system for addressing global localization problems.

However, there are still some difficulties to overcome in vision SLAM applications, such as loop closure detection. This issue concerns the difficulty of recognizing already mapped areas. It is an image retrieval task which determines whether query image has been taken from a known location. This task is similar with image classification methods. Generally speaking, the overall goal of the detection presented in this paper is to find the data collection node of already mapped areas which is closest to the query image.

A. Literature Review

From the literature, various methods have been proposed for visual loop closure detection. Local features matching is the basic method for detection. In this method, the scene in the image is described by multi-dimensional vectors. Compared with images, these vectors have ability to reduce data storage. Among the mass of descriptors, Scale-invariant Feature Transform (SIFT) is perhaps one of the most popular methods which was proposed in 2004 [6]. This descriptor provides the first approach to the problem of extraction stability. Various methods of SLAM have used SIFT to achieve local features matching. For example, Kosecka et al. [7] use SIFT to describe the scene in indoor environment. Then, a probabilistic environment model is set up to make location recognition. Similarly, Zhang et al. [8] also use SIFT to set up a bag-of-raw-features model, which is then used to realize visual loop-closure detection in autonomous robot navigation. However, SIFT descriptor suffers from low detection efficiency and complex computational processes. To solve this problem, Speeded-Up Robust Features (SURF) is proposed to simplify the computation complexity [9]. It has a similar extraction result to SIFT. Tongprasit et al. [10] adopt SURF descriptor to modify the Position-Invariant Robust Features (PIRF) method. The modified method is 12 times faster than the original method. However, the matching speed of SURF still cannot meet the request of loop closure detection in some situations, such as outdoor environment. Hence, another descriptor with fast matching speed is considered in our approach.

One of the disadvantages of local features matching is that local features still have huge data storage requirements due to their descriptors. To address this problem, the Bag-of-Words (BoW) algorithm is used for loop closure detection, which is

Y. Li is with the ITS Research Center, Wuhan University of Technology, Wuhan 430063, China, (e-mail: ycli@whut.edu.cn)

Prof. Z. Hu is with the ITS Research Center, Wuhan University of Technology, Wuhan 430063, China (corresponding author e-mail: zzhu@whut.edu.cn).

G. Huang is with the ITS Research Center, Wuhan University of Technology, Wuhan 430063, China, (e-mail: gh@whut.edu.cn)

Dr. Z. Li is with Department of Mechanical Engineering, Iowa State University, Ames 50010, America (e-mail: zhixiong.li@ieee.org).

Prof. M. Sotelo is with Department of Computer Engineering, University of Alcalá, Alcalá de Henares (Madrid), Spain (email: miguel.sotelo@uah.es)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

also based on local features extraction [11][12]. The algorithm used here treats the local features as visual words. Then, all the local features from a mapped area compose a visual dictionary. Furthermore, each image is represented by a histogram which is based on local features clustering. Since the histogram is more discriminative than the farther ones, the BoW algorithm has the ability to make location recognition. Compared to local features matching, this method can reduce the total sets of feature descriptors. There are also many studies using BoW method to achieve loop closure detection. For instance, Ho et al. [13] set up a multi-robot map-joining system to address the loop closure detection problem. The system uses BoW method to draw up an image classification scheme. Firstly, images from mapped areas are represented as visual words. The visual words from the dataset compose the visual dictionary. In the step of features matching, the authors use a Nearest Neighbor (NN) search to separate the corresponding histogram. The search results are used to represent the detection events. Similarly, in [14], the authors draw up a place recognition scheme which relies on the BoW method. A multiple-view algorithm is proposed to compute the ultimate results based on the matching results. In these two approaches, the use of BoW method improves the accuracy of matching, which is able to robustly deal with noisy images. The approaches discussed above mainly use multi-dimensional features. Cummins et al. also propose an appearance-based method for loop closure detection by using BoW method. Each visual word represents a high-dimensional feature descriptor. They use public datasets from Fast Appearance-Based Mapping (FAB-MAP) to evaluate their method and the method has good results [15]. Moreover, Galvez-Lopez et al. [16] attempt to create visual words from binary features. To enhance the robustness of matching, the hierarchical BoW model is used with key points which are detected by FAST method and described by BRIEF algorithm. They use sequences of about 19,000 images to detect the loop closures. However, in some scenes, the matching accuracy by using BoW method is not robust enough to meet the request of loop closure detection.

Another way to reduce data storage is holistic feature matching. When a whole image is considered as a feature, this feature is called holistic feature [17]. Holistic feature can describe the scene, which is used instead of image. The speed of holistic feature extraction is very fast. Hence, there are various detection methods using holistic feature matching. For example, Lategahn et al. [18] propose a holistic feature called an illumination robust descriptor (DIRD) to generate robust descriptors. In this descriptor, the authors use building blocks as describing objects. Hence, millions of descriptors can be used to construct this feature. Furthermore, a function is proposed to estimate DIRD descriptor, and loop closure detection is also presented in experiments. Similarly, Nourani-Vatani et al. [19] adopt the Optical Flow Moment (OFM) and the Optical Flow Shape Context (OFSC) as holistic feature descriptors. These descriptors are based on optical flow data to distinguish changes in scenes. To define each node of the dataset, the holistic features are extracted by statistical attributes from the optical flow. In the step of holistic feature matching, the Mahalanobis and χ^2 distance are computed between the query image and the dataset. The node with the least distance is obtained. In experiments, the proposed feature is evaluated in both indoor and outdoor environments, to prove that their feature can adapt different kinds of environment scenes. Moreover, Singh et al. [20] use GIST as holistic feature and extract these features from omni-directional images. To match the holistic features, the authors propose a new matching measure for the four views which the omni-directional image consists of. As a result, the loop closure detection is taken in an urban environment. Generally, to ensure the robustness of the holistic feature matching, some new descriptors are usually proposed in the studies. Thus, some new similarity measurements are also presented at the same time.

However, due to the massive data in the dataset, simply using feature matching can result in low detection accuracy and large computation cost. More advanced methods must be proposed in order to enhance the accuracy. Actually, the method for loop closure detection is strongly related to robot visual localization [21][22], as both of them include feature matching and place recognition. Hence, the methods for loop closure detection can be inspired by visual localization, as there are various advanced methods in visual localization. Ziegler et al. [23] set up an autonomous driving system which consists of global position detection based on visual features. They fuse features matching and 3D data registering. A 6D rigid-body transformation is computed and then the vehicle pose is found. They pick a 103-km route for real vehicle driving. The localization accuracy achieves centimeter-level by fusing the data of wheel encoders and yaw rate sensors. Besides, Wang et al. [24] present a coarse-to-fine localization method to divide the localization into two steps. Coarse localization is fast but not accurate enough, which can provide a set of possible locations. Then the accurate result is found in the candidate set. Besides, Son et al. [25] propose a key frame selection method to make coarse localization. Nodes from the dataset are divided into key frames and non-key frames by checking the matching number of feature points. In the step of feature matching, query image is first matched with key frames of the dataset. The closest key frame and their corresponding non-key frames are selected as coarse localization results. Topological model is another method to reduce the computation complexity and enhance the matching accuracy. For example, Lategahn et al. [26] take 30 positions as information to set up a topological model and a dynamic programming procedure is used to compute the node closest to the query image. They pick a 7-km route to present and estimate their method in urban environment. In addition, there are also some methods used in loop closure detection. For example, in [27] and [28], the authors use pose cells and local view codes to detect loop closure. The pose cells are composed of 3 degree of freedom (DoF) pose of the robot. The poses are collected by using a 3D version of continuous attractor networks. The 3D data have an ability to enhance the accuracy of loop closure. Moreover, in [14], before feature matching, the authors first set up a Bayesian filtering model to compute the probability of the query image and its previous images belonging to the same scene. Then, a scene with a sequence of nodes is picked for coarse detection. Furthermore, they adopt feature matching with the selected scene by using BoW method. As discussed above, coarse detection is crucial

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

for feature matching and place recognition. Hence, in our approach, we also propose a coarse detection method before accurate feature matching.

To enhance the detection accuracy, another method is storing three-dimensional (3D) data. 3D data can compute the ego-camera pose and the computed pose has an ability to check the detection results. Hence, various methods in the literature use 3D data, such as [4, 23, 26, 27, 28]. 3D data in these methods are usually obtained by laser scanner, binocular camera and RGB-D camera. However, both laser scanner and binocular camera are expensive. Such high cost will prevent the development of SLAM and vehicle localization. Although RGB-D camera has a low cost, its 3D data usually have a short range which is not suitable for outdoor environment. As mentioned above, this paper proposes a method for loop closure detection with low-cost sensors.

B. Contributions

In this paper, we propose an image sequence matching (ISM) method for loop closure detection. This method follows a three-step approach. At first, we set up a topological model which aims at making coarse detection to select a set of possible nodes from the already mapped areas. Furthermore, we take sequence matching by using both holistic feature and local features. Several candidates are selected in the two feature spaces. Last but not least, we fuse these candidates to find the closest node by using voting K-nearest neighbor (V-KNN) method. As a result, the closest node is found and the query image is updated to mapped areas. The contributions of this paper are summarized as follows:

1) A novel detection model called topological feature-zone is set up. In this model, we first propose feature-zone to distinguish different zones in mapped areas by using feature matching. In the step of loop closure detection, the query image is detected topologically with one previous result. One feature-zone is selected to predict the detection range by using this model. This model simplifies the detection procedure and increases the accuracy.

2) A novel image sequence matching method is proposed to improve the detection accuracy. This method matches the query image with several consecutive nodes. We select the middle one from the closest consecutive nodes as candidate. In the proposed method, the advantage is that image sequence matching can ensure detection stability and outliers can be eliminated in this step.

3) A novel clustering method for fusing candidates from different feature spaces, called voting-KNN (V-KNN) method, is proposed. The candidates are selected from different feature spaces such as holistic feature matching and local features with BoW. V-KNN votes all the candidates from different feature spaces and then selects one as result. The advantage is that if other candidates are introduced from other feature spaces, the V-KNN method can still provide the fusion of them.

The structure of this paper is organized as follows. Section I introduces the background and surveys the literature. Section II presents the ISM method for loop closure detection. Section III presents the experimental results with real data. Conclusions of this paper are drawn in Section IV.

II. THE IMAGE SEQUENCE MATCHING METHOD

The work presented in this paper is called image sequence matching (ISM) method. The illustration of the ISM method is shown in Fig. 1.

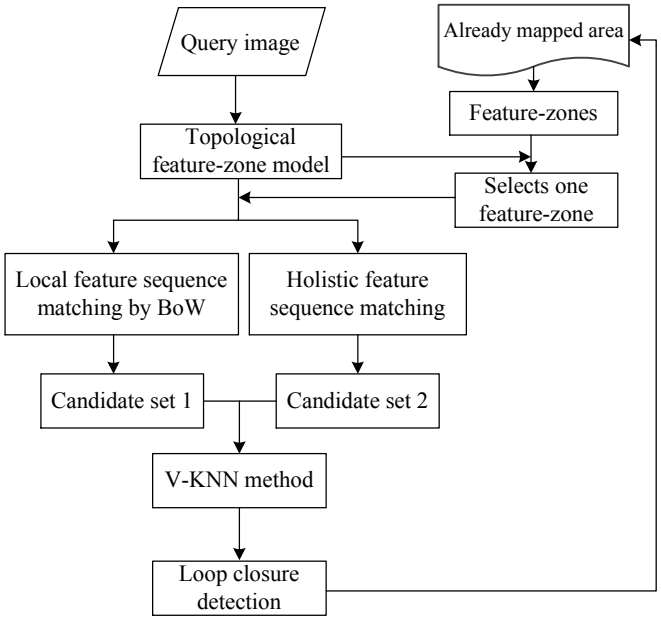


FIGURE 1. The proposed methodology for ISM loop closure detection

A. Holistic feature extraction

Before delving into the details of loop closure detection, we first extract holistic feature from each query image and each node. To enhance the extraction speed, we use ORB (oFAST and rBRIEF) descriptor for feature extraction in our method. ORB is a popular feature descriptor proposed in 2011[29]. This descriptor is very fast and has similar matching results compared with SIFT and SURF. The main idea is the combination of oFAST (FAST with orientation) and rBRIEF (rotated BRIEF).

More specifically, each holistic feature is described by BRIEF. The BRIEF descriptor is a bit string description of an image patch which is constructed by a sequence of binary intensity tests. To compute this descriptor, we denote s as a smoothed image patch. In patch s, two pixels t_1 and t_2 are performed a binary test by comparing their intensities. The formula is shown as follows:

$$\tau(s;t_1,t_2) := \begin{cases} 1, s(t_1) < s(t_2) \\ 0, s(t_1) \geq s(t_2) \end{cases} \quad (1)$$

where τ is a binary test; $s(t_i)$ is the intensity of image s at point t_i. The BRIEF descriptor is thus defined as a vector of n binary tests:

$$f_n(s) = \sum_{i \in [1,n]} 2^{i-1} \tau(s;t_{1i},t_{2i}) \quad (2)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

Many methods are shown in the literature, which are about how to choose n in the binary tests. In this paper, we select $n=256$ for vector length by using a Gaussian distribution around the patch center. Hence, we should resize each image into a standard ORB patch image before holistic feature extraction. The typical resolution of an ORB patch image is 63 × 63 pixels. As a result, the holistic feature descriptor for each image is represented with a 256-bit string, which is shown as Fig. 2.

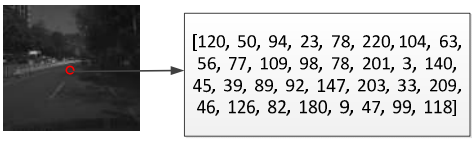


FIGURE 2. Extraction of ORB holistic feature. Image center as ORB feature point position and compute the corresponding ORB descriptor as holistic feature with each digit as a 8-bit char (totally 32×8bytes=256bits).

B. Topological Feature-Zone Model

The goal of loop closure detection is to find the node in mapped area which is closest to the query image. However, simply taking feature matching from the huge data source is error prone and susceptible to visual aliasing and ambiguities. In this step, we set up a topological feature-zone model for coarse detection.

In the already mapped area, we first compute the Hamming distance between two adjacent nodes. Hamming distance is computed by applying the XOR bit operation to two 256-bit strings of ORB holistic features as follows:

$$Hamm\left(X^{(1)},X^{(2)}\right)=\sum_{i}^{256}XOR\left(X_i^{(1)},X_i^{(2)}\right) \tag{3}$$

where $X^{(1)}$ and $X^{(2)}$ are two arbitrary adjacent holistic features. If the distance is below a threshold σ, they are considered as belonging to the same feature-zone. Otherwise, they belong to different feature-zones. In this way, the already mapped areas are divided into some feature-zones. Then, we need to select a feature-zone for the query image in the coarse detection stage.

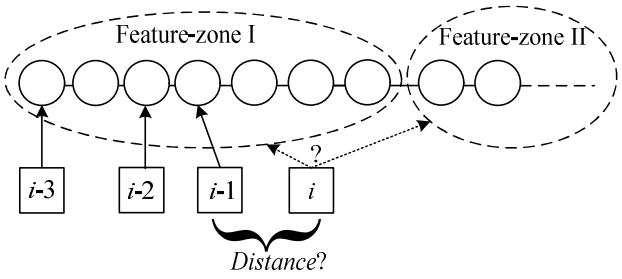


FIGURE 3. Illustration of topological feature-zone model. The squares denote query image, the circles denote nodes in already mapped areas.

To find the feature-zone, we set up a topological model. Similar to feature-zone classification, we compute the Hamming distance between the query image and its previous image. If the distance is below the threshold σ, we select the feature-zone which the previous image belongs to. Otherwise, we select the adjacent feature-zone. The topological feature-zone model is shown in Fig. 3. From this figure, the current query image is *i*, while its previous image is *i*-1. There are two feature-zones in this figure. Obviously, the previous image belongs to feature-zone I. If the distance between *i* and *i*-1 is below σ, we select feature-zone I as coarse detection result. On the contrary, if the distance exceeds this threshold, feature-zone II is selected as the detection result.

C. Image sequence matching

In this sub-section, the image sequence matching is used in both holistic feature space and local feature space to ensure the detection accuracy.

1) Holistic feature sequence matching

We have extracted the holistic feature of the query image and matched it with its previous image. In this step, we use sequence matching to match the query image with the selected feature-zone. The basic idea behind this approach is the following: if we want to select one node from the feature-zone as candidate, this node is not only similar to the query image, but also its adjacent nodes must exhibit a close distance. The advantage of sequence matching is that it can eliminate outliers and improve the matching stability.

More specifically, we consider that n (n=3, 5, 7…) arbitrary consecutive nodes from the feature-zone are organized into a site. Then the query image is compared with each site, i.e., Hamming distances are computed between the query image and each node of the site. As a result, the computed distances are accumulated in the following equation:

$$CHmatch\left(X^q,S\right)=\sum_{i=1}^{5}Hamm\left(X^q,X_i^{node}\right),X_i^{node}\in S \tag{4}$$

where X^q is the holistic feature of the query image; S is the matching site, while X_i^{node} is the holistic feature of *i*-th node in site S. In this step, we compute 5 closest sites and select the middle node of each site as candidates. Hence, there are 5 candidates which compose candidate set 1. The illustration of features sequence matching is shown in Fig. 4.

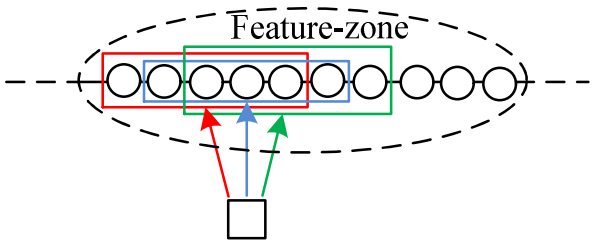


FIGURE 4. Illustration of features sequence matching: rectangles denote sites in feature-zone, square denotes query image and circles denote nodes.

2) Local feature sequence matching with BoW

Besides holistic features matching, we also extract local features by ORB descriptor to enhance loop closure detection. In this feature space, we use BoW method to realize local feature matching. Unlike holistic feature extraction, local features require to be detected at first. The features are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

detected by FAST, which measures the gray difference between the pixel of each point and its neighborhood. For each image, we first select an arbitrary point and compare its pixel with 16 pixels of its neighborhoods. If there are more than n consecutive neighborhoods, whose gray differences exceed a threshold, this point is treated as a local feature. In this study, we denote n=9. In this way, local features can be extracted in an automatic manner. The local features extracted from one image are shown in Fig. 5. Each feature is also represented with a 256-bit string.

FIGURE 5. Extraction of ORB local features.

Furthermore, the implementation of the BoW method used here is to reduce the data storage. The method is followed by a two-step approach. First of all, each feature-zone is selected as a visual dictionary. All the local features in this zone are clustered by using K-means clustering method [30]. In this method, Hamming distance is computed to measure the distance between two local features. Then, each center of cluster can be computed by this distance and these centers compose visual words of the dictionary. As a result, we obtain a set of visual words for each feature-zone, which is denoted by $\{v_w^1, v_w^2, v_w^3, \ldots, v_w^n, \ldots\}$. In the second step, each local feature of the query image is compared to the visual word with the least Hamming distance. Then we can derive a histogram over visual words for each query image. The histogram is shown in Fig. 6. Each histogram is represented as follows:

$$H = [p_1, p_2, \ldots p_n] \tag{5}$$

where p_i is the frequency of the i-th visual word and there is a total of n visual words. Hence, in the step of local features sequence matching, we compute Euclidean distance between the query image and each node. As the histogram denotes the frequency of visual word, Euclidean distance is suitable for similarity computation of frequency. Then we sum the distance of n (n=3, 5, 7…) consecutive nodes. The distances of sequence matching is computed as follows:

$$CLmatch(H^q, L) = \sum_{i=1}^{5} Euc(H^q, H_i^{node}), H_i^{node} \in L \tag{6}$$

where H^q is the histogram of the query image; L is the local features of matching site, while H_i^{node} is the histogram of i-th node in site L. The formula of Euc () is computed as follows:

$$Euc(H^1, H^2) = \sqrt{\sum_{i=1}^{M} (p_i^1 - p_i^2)^2}, \quad p_i^j \in H^j \tag{7}$$

In this step, we also compute 5 closest sites and select the middle node of each site as candidates. Therefore, the 5 candidates compose the candidate set 2. As a result, we obtain two candidate sets from different data spaces. In the next step, we will select one candidate as the detection result.

(a) (b)

FIGURE 6. Result of BoW: (a) histogram with BoW; (b) image for BoW

D. Loop closure detection with V-KNN

We have got 2 candidate sets from holistic feature space and local feature space, respectively. The detection result is included in these two sets. Hence, we propose a new cluster method called voting-KNN to fuse the sets and compute the detection result.

First of all, we define each different candidate from one candidate set as a "type". Therefore, the 10 candidates from two sets may have less than 10 types. Furthermore, these types from different feature spaces are fused in a box, which is shown in Fig. 7. From the figure, there are 8 types from "A" to "H". In these types, A and B are both selected twice while the others are selected only once. At this stage, we do not know which type is the detection result since two types are selected twice. Hence, we propose a voting system to solve this problem.

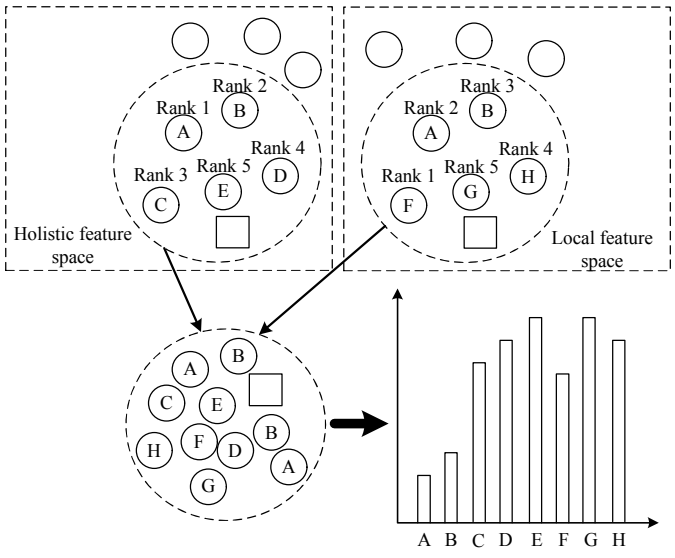


FIGURE 7. Illustration of V-KNN for loop closure detection. Squares denote query image. Circles denote candidates. Letters from "A" to "H" denote types.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

In each feature space, the candidates are ranked by computing Hamming distance and Euclidean distance, respectively. The candidate with least distance is ranked 1 while the one with largest distance is ranked 5. Then, we vote for each type based on their rank. The votes of the type are the sum of its ranks in two feature spaces. If the type is not selected in one feature space, we define that its rank is 10 in this feature space. For instance, the rank of type A is 1 in holistic feature space and 2 in local feature space in Fig. 7. As a result, the votes of A is 3. However, type C is only selected in holistic feature space and its rank in this feature space is 3. The votes of C is 13. All the types are voted and derive a histogram which is shown in the sub-figure of Fig. 7. We select the type with the least votes as detection result. If there are more than two types with the same votes, we select the type with less Euclidean distance.

E. Outline of the ISM Algorithms

The algorithms for ISM can be summarized as follows:

1) We extract holistic features for query image and nodes. In the already mapped area, each pair of two adjacent nodes is computed for Hamming distance. Then, the mapped area is divided into several feature-zones.

2) In the step of loop closure detection, we detect the first query image as prediction information. Next, for each query image, we compute the Hamming distance between the query image and its previous image. Then we select a feature-zone for coarse detection.

3) The query image is matched with the consecutive nodes from the feature-zone in different feature spaces. In each space, 5 candidates are selected to compose a candidate set.

4) V-KNN method is used to fuse the two candidate sets. The type with the least votes is selected as the detection result.

5) We utilize local feature matching to check each result. If the number of matching points are lower than a threshold σ=45, we treat it as an outlier. As a result, the inliers are updated to the corresponding feature-zone. The outliers are reconstructed the feature zones again in the offline training phase.

III. EXPERIMENTAL RESULTS

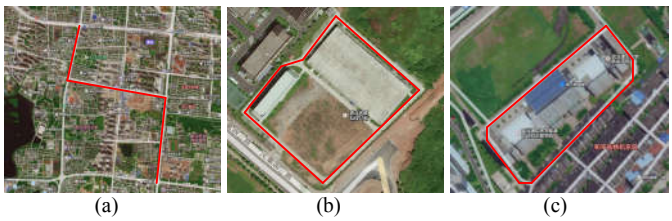
Next, we present experiments with collection datasets and public datasets to evaluate and assess our method. On the collection datasets, we picked 3 different routes in Wuhan City, China. These routes were traveled at different times and covered different weather conditions. On the public datasets, we use FAB-MAP datasets which include 2 datasets collected from City Centre and New College.

A. Loop closure detection in different scenes on collection datasets

To collect the datasets, two vehicles with monocular cameras were provided. One of them was a standard vehicle with a forward camera. The camera was an rs-2300-gc camera produced from Beijing Microview Company and each image taken by this camera had a size of 1600×1200 (in pixel). Another one was a trolley with a forward smartphone. This smartphone had an IMAX 333 camera produced from Sony Company and each image taken by this camera had a size of 800×600 (in pixel). Fig. 8 shows the setup of data collection system, the white circle show the cameras in the vehicles. Then, we used these data collection vehicles to make loop closure detection in different scenes, at different times and in different weather conditions.

FIGURE 8. Setup of data collection system

The 3 different routes covered various scenarios such as industrial park, urban roadway and campus. To collect data in the first two routes, we used the standard vehicle. The trolley was used to accomplish data collection in campus. These test sites are shown in Fig. 9. The total length of these routes was over 10 km. We collected data twice for each test route. Dataset was set up in the first time collection. The data collection frequency was no less than 2 meters/frame. Thereafter, the images for loop closure detection were collected in the second time.



(a) (b) (c)

FIGURE 9. Test routes: (a) urban roadway; (b) industrial park; (c) campus

For each collection image, we first made pre-processing such as image gray processing and histogram equalization. Thereafter, we resized these images into standard images with the resolution of 63 × 63 (pixel) for holistic feature extraction. The ORB descriptor for each holistic feature was represented with 256-bit string. Furthermore, the dataset was divided into a sequence of feature-zones by computing Hamming distance. Last but not least, we used the ISM method to make loop closure detection. For each detection result, we utilized local feature matching to match the result with its query image. We selected a threshold σ=45. If the number of matching points were lower than the threshold, the result was treated as an outlier. The advantage was that some bad detection results could be removed in this way.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

To select a parameter n for sequence matching, we select 10 query images in each route. Then, we select different parameters such as n=3, 5, 7, 9 to compute detection accuracy. The test results are shown in Fig. 10. From this figure, we use frame as unit to evaluate our method. This unit means the frame difference between the test result and the ground truth data. The ground truth data are computed manually. We can see that the results perform best when we set n=3. Therefore, we select n=3 in the next tests.

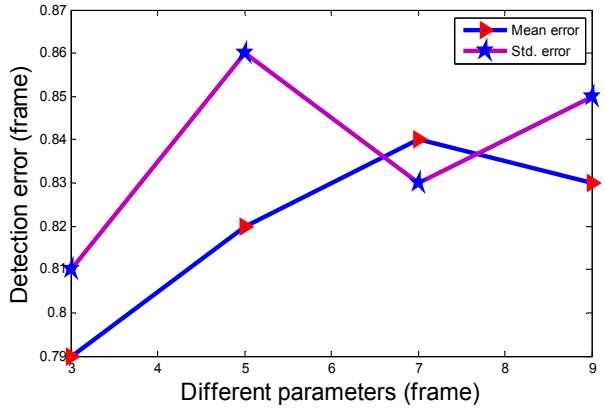


FIGURE 10. Detection results in different parameters

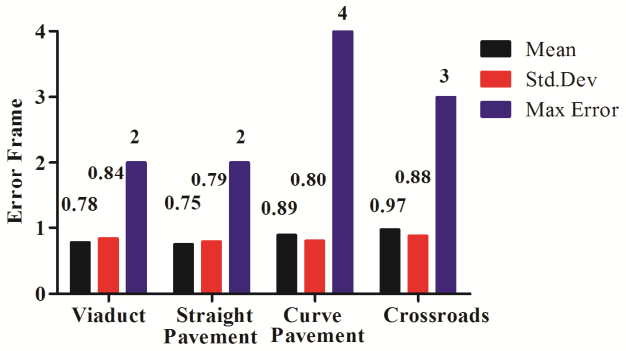


FIGURE 11. Detection results in different scenes

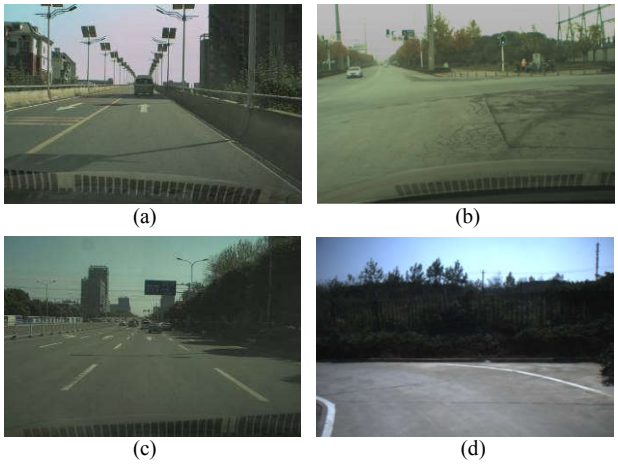


FIGURE 12. Various situations of dataset: (a) viaduct; (b) crossroads; (c) straight pavement; (d) curve pavement

Fig. 11 shows the detection results of the 3 routes. To demonstrate the detection result, we divided the 3 routes into different scenes in this figure, such as crossroad, curve pavement, straight pavement and viaduct. Fig. 12 gives examples of images from different scenes. From Fig. 11, as the view angle has a great change in the scenes of crossroads and curve pavement, the results in these 2 scenes are worse than the results that in another 2 scenes, both in terms of mean error and max error. Fortunately, even though max errors in crossroads and curve pavement reach 3 frames and 4 frames, respectively, the means and standard deviations of 4 scenes are all less than 1 frame. These results show that the proposed method has high accuracy and stability. Furthermore, what happens when the proposed method is compared to other methods?

To evaluate our method, we adopt two previous methods for comparison, which are the method in [18] and the method in [25]. The comparison results are shown in TABLE I. In each test of the scene, we select 100 images for viaduct, 50 images for crossroads, 150 images for straight pavement and 80 images for curve pavement, respectively. As the frequency of each scene is different in the real road environment, we select different image numbers for each scene. From the table, we define that if the error frame is no more than 1 frame, it is a correct detection. It is because the image collection frequency is no less than 2 m/frame in the first round collection. The distance between 2 adjacent nodes are very close.

TABLE I Comparison of results from the ISM method, the method in [18] and [25] in different scenes

Scenes	#Image	Accuracy by ISM	Accuracy by the method in [18]	Accuracy by the method in [25]
Viaduct	100	91.0%	88.0%	81.0%
Crossroads	50	86.0%	82.0%	70.0%
Straight Pavement	150	94.6%	86.0%	84.0%
Curve Pavement	80	87.5%	81.3%	77.5%

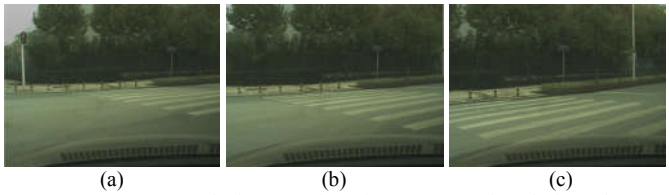


FIGURE 13. An example for error detection: (a) ground truth data; (b) query image; (c) test result

From the results we find that the ISM method performs better in each scene than the method in [18] and [25]. The accuracies by ISM in all scenes are more than 85%. In viaduct and straight pavement, ISM performs high accuracies which are more than 90%. All the statistics show that ISM method has high accuracy and great robustness. However, we can also find that the accuracies in crossroads and curve pavement are also lower than the accuracies that in the other two scenes. Herein, we give an example for error detection in the scene of crossroads, which is shown in Fig. 13.

Fig. 13 (a) is the node closest to the query image (shown in Fig. 13 (b)). However, the direction of vehicle changes greatly. The signal light has disappeared in the next position.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

Therefore, the similarity between the query image and Fig. 13 (c) is greater than that of the ground truth. As a result, Fig. 13 (c) is selected as the test result.

B Loop closure detection at different times on collection datasets

We have shown the detection results in different scenes. Then, what would happen if the tests were conducted at different times? Hence, in this sub-section, we introduce the performance of the ISM method at different times.

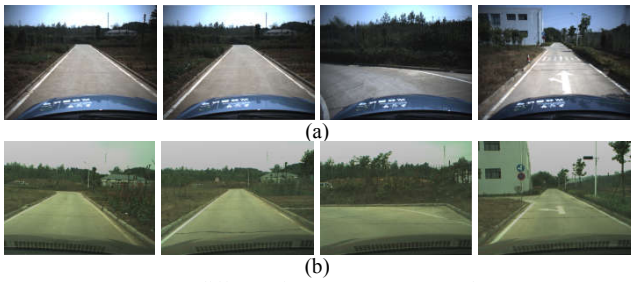


FIGURE 14. Test routes at different times: (a) 10:00 a. m. (b) 5:00 p. m.

We also carried the tests at two different times, 10:00 a. m. and 5:00 p. m. The reason why we selected these 2 times was that the sunlight was strong at 10 o' clock, while it would be sunset at 5:00 p. m. As shown in Fig. 14, the illuminations were quite different in these 2 times even though they were both collected in a sunny day. To evaluate our method, we selected 150 images at each time. These images also included several scenes such as straight pavement, crossroads and curve pavement. The detection results are shown in Fig. 15.

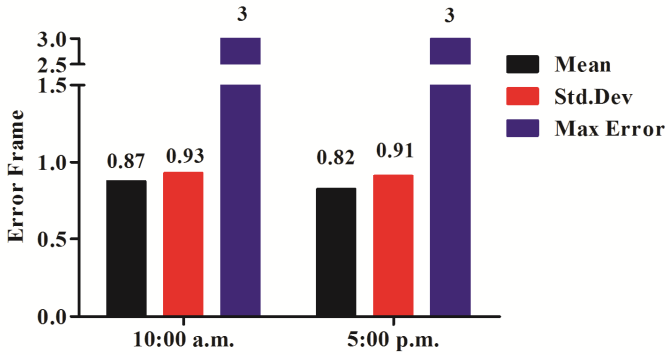


FIGURE 15. Detection results at different times

TABLE II Comparison of results from the ISM method, the method in [18] and [25] at different times

Times	#Image	Accuracy by ISM	Accuracy by the method in [18]	Accuracy by the method in [25]
10:00 a.m.	150	91.1%	83.3%	75.3%
5:00 p.m.	150	92.3%	85.1%	79.3%

From Fig. 15, the detection results have a bit change whether in mean or in standard deviation and they have the same max error at different times. Specifically, the mean errors at different times are both less than 1 frame. It means that our method has a great robustness in illumination changes. Moreover, we also compare the ISM method with the method in [18] and [25], which is shown in TABLE II. From this table, we can find that the proposed method performs better than another two methods do. In addition, the accuracy change is from 92.3% to 91.1%. The data of accuracy change is also less than that in another two methods.

C Loop closure detection in different weather conditions on collection datasets

Next, we carry out detection tests during a rainy day and a sunny day to evaluate our method in different weather conditions. The test routes are shown in Fig. 16. From Fig. 16 we can find that the pavement was wet. Thus there were not only illumination changes in the scene, but also scene of background changes. Hence, it is a challenging task for our method to make loop closure detection.

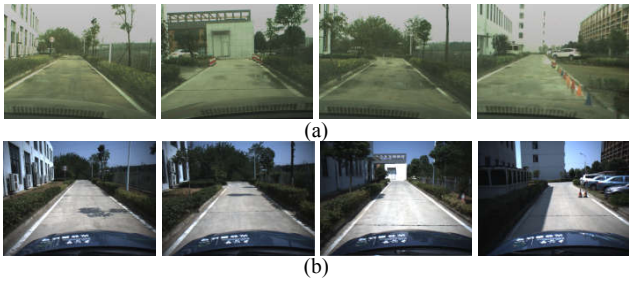


FIGURE 16. Test routes in different weather conditions: (a) rainy day (b) sunny day

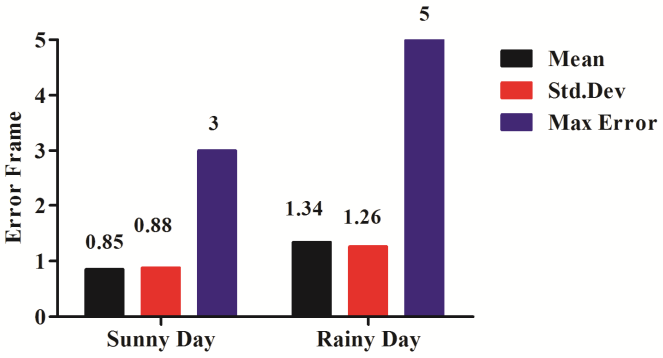


FIGURE 17. Detection results in different weather conditions

TABLE III Comparison of results from the ISM method, the method in [18] and [25] in different weather conditions

Weather conditions	#Image	Accuracy by ISM	Accuracy by the method in [18]	Accuracy by the method in [25]
Sunny day	150	91.7%	84.6%	77.3%
Rainy day	150	78.9%	70.6%	61.3%

Herein, we also provide a figure to show the detection results and set up a table for results comparison. They can be shown in Fig. 17 and TABLE III. In this test, 150 images were selected in different weather conditions. From Fig. 17, compared with the results in sunny day, we can find that the results have a moderate decrease in rainy day test. The mean error is more than 1 frame and the max error reaches 5 frames. Besides, the rainy day test also has lower detection accuracy than the sunny day test. Moreover, compared with other methods, the ISM method also exhibits better performance and a higher detection results. Fortunately, the rain was not heavy on that day and this rainy day did not cause a huge effect for detection. Although the accuracy has dropped, the detection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2725387, IEEE Access

accuracy in that day is still 78.9%. This accuracy of our method is higher than that in [25] on a sunny day.

D. Result with FAB-MAP Data Sets

The proposed method was also tested by the public data sets from FAB-MAP [15, 28]. This datasets are sequence images containing two datasets which are City Centre and New College collected in the UK. Each dataset includes sequence images, image collection coordinates (GPS), ground truth, aerial photo, etc. The scenarios in different datasets are shown in Fig. 18.

(a)

(b)

FIGURE 18. Scenarios on different datasets: (a) dataset on City Centre; (b) dataset on New College

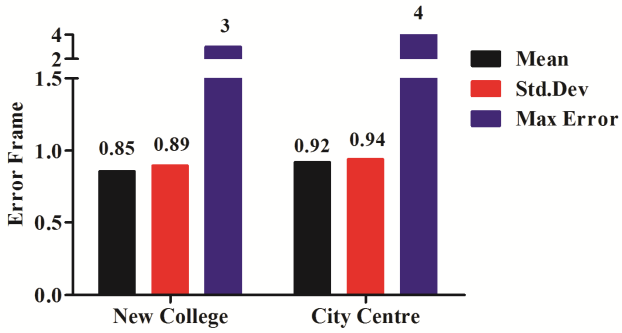


FIGURE 19. Detection results on the FAB-MAP datasets

On each dataset, we selected 200 images for query images, respectively. The others were used as training data. Fig. 19 shows the detection results on the 2 datasets. From this figure, the results on the New College dataset perform better than that of on the City Centre dataset. The mean errors on the two different datasets are all lower than 1 frame. Overall, the detection results on FAB-MAP datasets are similar with the results on the collection datasets. They mean that our method has a great robustness on different datasets. Moreover, we also compare the proposed method with the method in [18] and [25], which is shown in TABLE IV. From this table, we can find that the ISM method performs better than that another two methods do whether on City Centre or on New College.

TABLE IV Comparison of results from the ISM method, the method in [18] and [25] in FAB-MAP dataset

Dataset	#Image	Accuracy by ISM	Accuracy by the method in [18]	Accuracy by the method in [25]
City Centre	200	88.9%	86.8%	81.4%
New College	200	92.3%	87.5%	82.1%

E. Further evaluation

As we have used the number of local features matching as threshold, some detection results are removed in the above experiments. However, although adjusting the threshold can obtain high detection accuracy, the detection rate would decrease at the same time. Hence, we use precision recall curves to evaluate the ISM method.

The precision recall curves are shown in Fig. 20. The curves were generated by varying the threshold. We took 5 routes from the collection datasets and the FAB-MAP datasets, which were industrial park, campus, urban roadway, New College and City Centre. "Recall" here is the proportion of the number of inliers to total number of detections. From this figure, when the precision rates achieve 100%, the method has the recalls at 57%, 53%, 38%, 48% and 41% for each dataset. The vehicles in urban roadway and City Centre affect the recalls at 100% precision.

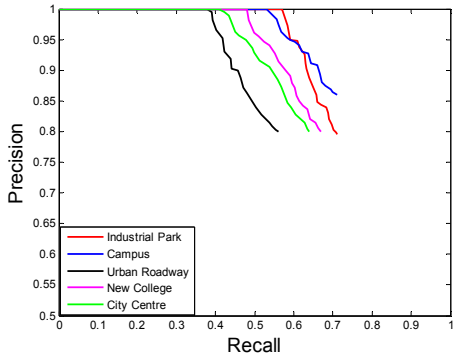


FIGURE 20. Precision-recall curves for different datasets

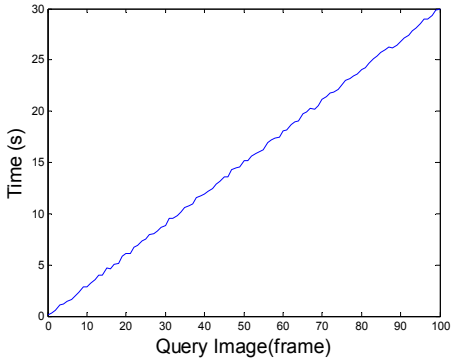


FIGURE 21. Processing time for query images

Moreover, we also evaluated the ISM method in the terms of efficiency. First of all, we detected the processing time to generate all the feature-zones. As the feature-zones were divided by holistic feature matching, the processing time was millisecond-level for one training image. We selected 1500 images as training data to generate feature-zones. The processing time was about 20 s and all the processing of generation was offline. The number of generated feature-zones was 17. Furthermore, we evaluated the running time for each query image. The running time for query images is shown in Fig. 21. From this figure, we selected 100 query images. In the step of feature-zone selection, query images were only matched with their previous images and then they selected the

feature-zone topologically. In the step of detection, the processing included both holistic feature matching and local feature matching with BoW. As a result, the running time for each query image was about 0.3 s. The total of running time was about 30 s.

IV. Conclusions

The solution proposed in this paper is realized by image sequence matching method. In this method, feature-zone is proposed to set up a topological model and the model used here provides a simple way to reduce the feature matching range. Then, 2 kinds of feature matching are used to enhance the detection accuracy, such as holistic feature matching and local features with BoW. Meanwhile, the use of these 2 kinds of features can reduce the data storage on the dataset. In addition, the proposed feature sequence matching method also has an ability to improve the accuracy. Last but not least, a novel clustering method called V-KNN is proposed to fuse candidates in different feature spaces. This method does not require any additional sensors and only needs a low-cost monocular camera even though a smartphone. As a result, the experimental results presented here show that the proposed method has a great robustness on different datasets and illumination changes. In these situations, the mean of detection errors are all less than 1 frame and the accuracies are more than 90%. Although there is a moderate drop in the rainy day test, the proposed method also exhibits higher accuracy compared with other methods. In the future work, we will focus on the detection in rainy condition and the detection accuracy will be improved in different weather conditions.

Acknowledgements

The work presented in this paper was funded by the Major Project of Technological Innovation in Hubei Province (No. 2016AAA007), National Natural Science Foundation of China (No. 51679181) and the Science-technology Funds for Overseas Chinese Talents of Hubei Province (No. 2016-12). We also appreciate Wuhan Kotei Informatics Company for the assistance in experiment setup and data collection.

References

[1] K. Pahlavan, P. Krishnamurthy and Y. Geng. “Localization challenges for the emergence of the smart world,” *IEEE Access*, vol. 3, pp. 3058-3067, 2015.

[2] D. Gálvez-López, M. Salas, D. Tardós, et al. “Real-time monocular object slam,” *Robot. Auton. Syst.* vol. 75, pp. 435-449, 2016.

[3] T. Dharmasiri, V. Lui and T. Drummond. “MO-SLAM: Multi object SLAM with run-time object discovery through duplicates,” *IEEE Int. Conf. Intel. Robot. Syst. (IROS)*, 2016, pp. 1214-1221.

[4] T. Whelan, M. Kaess, H. Johannsson, et al. “Real-time large-scale dense RGB-D SLAM with volumetric fusion,” *Int. J. Robot. Res.*, vol. 34, no. 4, pp. 598-626, 2015.

[5] H. Chen, D. Sun and J. Yang. “Global localization of multirobot formations using ceiling vision SLAM strategy,” *Mechatronics*, vol. 19, no. 5, pp. 617-628, 2009.

[6] G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[7] J. Kosecka and X. Yang. “Location recognition and global localization based on scale invariant features,” *in: European Conference on Computer Vision*. 2004.

[8] H. Zhang. “BoRF: Loop-closure detection with scale invariant visual features,” *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011, pp. 3125-3130.

[9] H. Bay, A. Ess, T. Tuytelaars, et al. “Speeded-up robust features (SURF),” *Comput. Vis. Image Und.*vol. 110, no. 3, pp. 346-359, 2008.

[10] N. Tongprasit, A. Kawewong and O. Hasegawa. “PIRF-Nav 2: Speeded-up online and incremental appearance-based SLAM in an indoor environment,” *IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 145-152.

[11] E. Nilsback and A. Zisserman. “A visual vocabulary for flower classification,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1447-1454.

[12] G. Csurka, C. Dance, L. Fan, et al. “Visual categorization with bags of keypoints,” *European Workshop on statistical learning in computer vision (ECCV)*, 2004, pp. 1-22.

[13] L. Ho and P. Newman. “Detecting loop closure with scene sequences,” *Int. J. Comput. Vision*, vol. 74, no. 3, pp. 261-286, 2007.

[14] A. Angeli, D. Filliat, S. Doncieux, et al. “A Fast and Incremental Method For Loop-closure Detection Using Bags of Visual Words,” *IEEE Trans. Robot*, *Special issue on Visual SLAM*, pp: 1-11, 2008.

[15] C. Mark, and P. Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance." *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647-665, 2008.

[16] D. Galvez-Lopez, D. Tardos. “Real-time loop detection with bags of binary words,” *IEEE Int. Conf. Intel. Robot. Syst. (IROS)*, 2011, pp. 51-58.

[17] J. Wu and M. Rehg. “CENTRIST: A visual descriptor for scene categorization,” *IEEE trans Pattern Anal. Mach. Intel*, vol. 33, no. 8, pp. 1489-1501, 2011.

[18] H. Lategahn, J. Beck, B. Kitt, et al. “How to learn an illumination robust image feature for place recognition,” *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 285-291.

[19] N. Nourani-Vatani, P. Vinicius, K. Borges and et al. “On the use of optical flow for scene change detection and description,” *J. Intell. Robot. Syst.* 2014, vol. 74, no. 4, pp. 817-846.

[20] G. Singh and J. Kosecka. “Visual loop closing using gist descriptors in manhattan world,” *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010, 4042-4047.

[21] H. Badino, D. Huber and T. Kanade. “Real-time topometric localization,” *IEEE Int. Conf. Robot. Autom. (ICRA).* 2012, pp. 1635-1642.

[22] H. Badino, D. Huber and T. Kanade. “Visual topometric localization,” *IEEE Intelligent Vehicles Symposium, IV*, 2011, pp. 794-799.

[23] J. Ziegler, P. Bender, M. Schreiber, et al. “Making Bertha drive—An autonomous journey on a historic route,” *IEEE Intell. Trans. Syst. Magazine*, vol. 6, no. 2, pp. 8-20, 2014.

[24] J. Wang, H. Zha and R. Cipolla. “Coarse-to-fine vision-based localization by indexing scale-invariant features,” *IEEE Trans. Syst. Man Cybern*, vol. 36, no. 2, pp. 413-422, 2006.

[25] J. Son, S. Kim and K. Sohn. “A multi-vision sensor-based fast localization system with image matching for challenging outdoor environments,” *Expert Syst Appl*, vol. 42, no. 22, pp. 8830-8839, 2015.

[26] H. Lategahn and C. Stiller. “Vision-only localization,” *IEEE Trans. Intell. Transp.*, vol. 15, no. 3, pp. 1246-1257, 2014.

[27] M. Milford and G. Wyeth. "Mapping a suburb with a single camera using a biologically inspired SLAM system." *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1038-1053, 2008.

[28] A. Glover, W. Maddern, M. Milford, et al. "FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day." *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2010, pp. 3507-3512.

[29] R. Mur-Artal, J. Montiel and D. Tardós. “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Trans. Robot.* vol. 31, no. 5, pp. 1147-1163, 2015.

[30] T. Kanungo, M. Mount, S. Netanyahu, et al. “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE trans Pattern Anal. Mach. Intel*, vol. 24, no. 7, pp. 881-892, 2002.