IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 18, 2019, accepted April 30, 2019, date of publication May 6, 2019, date of current version May 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914943

Manifold Regularization Graph Structure Auto-Encoder to Detect Loop Closure for Visual SLAM

ZHONGHUA WANG[1,2], ZHEN PENG[3], YONG GUAN[1,2], (Member, IEEE), AND LIFENG WU[1,2], (Member, IEEE)

[1]College of Information Engineering, Capital Normal University, Beijing 100048, China
[2]Beijing Key Laboratory of Electronic System Reliable Technology, Capital Normal University, Beijing 100048, China
[3]Information Management Department, Beijing Institute of Petrochemical Technology, Beijing 10217, China

Corresponding author: Lifeng Wu (wulifeng@cnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873175, Grant 71601022, and Grant 61472468, in part by the Natural Science Foundation of Beijing under Grant 4173074, in part by the Key Project B Class of Beijing Natural Science Fund under Grant KZ201710028028, and in part by the Youth Innovative Research Team, Capital Normal University.

ABSTRACT Loop closure detection plays a vital role in the visual simultaneous localization and mapping (SLAM) systems. In order to overcome the shortcomings of the artificial design algorithm to extract insufficient features, this paper proposes a graph-regularization stacked denoising auto-encoder (G-SDAE) network that achieves high detection accuracy and improve reliability. This method is based on the SDAE and the manifold learning graph regularization structure. The G-SDAE preserves the local abstract geometry structure between features through spatial mapping in manifold learning, and the G-SDAE network can automatically extract abstract features, avoiding relying on empirical design algorithms to extract low-quality visual features. Compared with the bag-of-words (BoW) method, the OpenFABMAP algorithm, and the traditional SDAE method, extensive experiments show that the proposed algorithm achieves superior performances and provides a feasible solution for the loop closure detection part of the visual SLAM.

INDEX TERMS SLAM, stacked denoising auto-encoder, loop closure, manifold learning.

I. INTRODUCTION

The concept of SLAM was first proposed by Smith, Self and Cheeseman in 1986 [1]. The SLAM is considered to be one of the key technologies in the research of autonomous robots, and is the key to realizing truly autonomous mobile robots [2]. The core of the SLAM problem is to locate itself based on location estimates and maps, and build incremental maps based on its own positioning to achieve robotic autonomous positioning and navigation [3]. The complete visual SLAM system relies on several key parts: i) the Visual Odometry (VO) estimate camera motion between adjacent images [4]; ii) optimizing feedback from camera pose and loop closure detection [5]; iii) loop closure detection determines whether the robot returns to the position that has been reached before [6]; iv) building a globally consistent trajectory and visually dense maps, etc. [7].

If only relying on the camera pose calculated by the Visual Odometry to consider the motion of the robot in the adjacent time, it will accumulate a large amount of motion error to make the SLAM system unreliable and SLAM system cannot construct globally consistent motion trajectories and maps [8]. Loop closure detection provides the correlation between current and historical data. Therefore, the robot can use the feedback information to reposition itself or help map and registration algorithms to obtain more accurate and consistent results [9]. The methods applied to loop closure detection are mainly divided into three ideas [10]: i) The detection methods based on topological maps; ii) The detection methods based on traditional appearance; iii) The detection methods based on deep learning. We will introduce the relevant methods based on the above three ideas.

First, the most specific mapping framework based on the mapping relationship between topological maps is topology metric or topological mapping [11]. Burgard et al. built independent local maps in real time based on a layered graph approach, and data association techniques stitch these matching maps together to automatically detect loop closures [12]. Latif et al. used topological information to represent position recognition as a sparse convex L1 minimization problem, and

The associate editor coordinating the review of this manuscript and approving it for publication was Chenping Hou.

applied effective homotopy methods to provide loop closure assumptions [13]. Since image retrieval can be enhanced by adding topological metric information, we can enhance the topology by adding some metrics such as distance and direction. For example, FAB-MAP [14] and SeqSLAM [15] originally used a purely topological system for loop closure detection, and CAT-SLAM found that using local metric pose filtering to enhance position recognition based on sequential appearance can improve the reliability of FAB-MAP loop detection method [16]. The SMART improved the general applicability of SeqSLAM by integrating spatial metric information to form spatially consistent sequences and new image matching techniques [17]. However, because sparse mapping based on topological relationships typically contains too little useful information to effectively eliminate interference information and correctly determine the loop closure trajectory.

Secondly, the appearance of images can provide rich information for loop closure detection. Therefore, loop closure detection methods based on traditional appearance appeared [18] and successfully applied to various practical SLAM systems [19], [20]. The traditional appearance features mainly include local feature descriptors such as Scale-Invariant Feature Transform SIFT [21] and Speed Up Robust Feature SURF [22], and global Gist [23] features that do not have a detection phase but need to process the entire image. Many advanced appearance-based loop closure detection algorithms employ the Bag-of-Words (BoW) model to quantify local features into vocabularies to improve efficiency [14], [24]. Usually, the image frames are assigned an image-visual-word-vector (descriptor) generated by local feature detection methods such as SIFT or SURF. The vector corresponds to the weighted appearance frequency of each visual word in a given image (histogram), clustering the visual feature descriptors into a collection of ''dictionaries'' [25]. This dictionary contain the centers of clusters considered to be independent of each other, by calculating these similarity between vectors to identify loop closure trajectories [18]. Cummins et al. proposed a new BoW method, based on tree structure accelerate the operation efficiency [26], [27]. The Gist extracts information from the image using Gabor filters of different directions and frequencies, and averages the results to generate a compact vector representing the ''points'' of the scene for loop closure detection and relocation [28], [29]. However, the BoW method still has its limitations. First, as the amount of data collected by the sensor increases, a huge dictionary needs to be constructed. In a very similar scene, there is a great ambiguity in the dictionary constructed by extracting the word histogram, which leads to an increase in the probability of false positive rate [30]. Second, the visual features are usually manually designed by the researchers in computer vision area, but each feature has its own uniqueness and complexity. Third, the criteria of designing traditional visual features is the feeling of humans, i.e., the features are expected to be stable under the change of the camera view or distance. However, this cannot be always guaranteed during the dynamic process.

Finally, in recent years, deep learning has shown strong advantages in various computer research fields, such as image recognition, classification, understanding, etc. It provides an opportunity for us to solve the loop closure detection problem in SLAM. The deep learning approach attempts to learn representative features directly from raw image data through a multi-layer neural network [31], which will be used to classify or identify tasks. The most relevant to our work is the unsupervised training deep network based on SDAE [32]. It performs image continuity hypothesis and sparse constraint and applies it in the scene of SLAM loop closure detection [33]. At the same time, the CNN network as a feature extractor can learn related general features in different visual tasks [34], [35]. Hou et al. used the intermediate layer of the Convolutional Neural Network (CNN) to construct feature descriptors. The output of each layer is treated as a feature vector by normalization, which represents an abstract representation of the input layer for loop closure detection [36]. Sünderhauf et al. comprehensively evaluated and compared the effectiveness of three state-of-the-art ConvNets on robot navigation related issues. They found that intermediate features exhibited robustness to changes in appearance, while advanced features carried more semantic information that could be used to partition the search space [37]. However, the above mentioned networks only focus on the global feature information of the original image data and, ignoring the spatial local geometric association between the image features. In particular, the pooling layer of the CNN also destroys this local geometric relationship [38].

Based on the graph regularization auto-encoder to achieve image classification work [39]. We improve the loss function of the network and the construction method of the original undirected graph of the local geometry. Then, we propose a network (G-SDAE) based on joint training of graph regularization auto-encoder and SDAE. The graph regularization structure in the G-SDAE network can learn the abstract local geometry between features in the reconstructed spatial geometric matrix process. The whole network training process mainly include independent pre-training, independent fine-tuning and joint fine-tuning. The similarity between key-frames is calculated by match these features, and further detect the loop closure within a certain similarity range. We discusses the effects of visualization Precision-Recall (P-R) curves and Average-Precision (A-P) scores under various methods. It can be seen that the G-SDAE network has a good performance in each dataset. In short, our main contributions are:

1) We propose an improved unsupervised deep network based on manifold learning and auto-encoder combination. It can effectively learn the local geometry structure of embedded subspaces and extract features.

2) G-SDAE network is applied to loop closure detection in visual SLAM, overcoming the shortcomings of the traditional algorithm to extract insufficient features and achieves high detection accuracy.

3) We design a set of algorithms to generate the ground truth of the datasets so that consistent evaluation criteria can be used for all relevant datasets to eliminate errors generated by different GPS sensors.

II. LEARNING FEATURE EXPRESSION

The most important aspect of detect loop closure in a appearance-based method is how to extract abstract features from the image that can express the core information of the original data. This process is called coding in auto-encoder network. For example, giving a raw image data containing n patches of images $X = [x_1, ..., x_n]^T \in R^{n\times m}$, each image patch is pulled into a m-dimensional vector. The purpose of coding is to reduce the dimensionality while learning features with stronger representation ability from the hidden layer $H = [h_1, ..., h_n]^T \in R^{n\times k}, k < m$. Manifold learning aims to enable low-dimensional data to reflect certain essential structural features of high-dimensional data [40]. Some high-dimensional data is actually a low-dimensional manifold structure embedded in high-dimensional space. For example, manifold M is a true subset $M \in R^N$ of N-dimensional European space (Euclidean space is a linear space with inner product, and the metric of geometric space is generalized in linear space). So we can achieve $M \to R^K$ mapping relationship by Manifold learning, that is mapping high dimensional data into low dimensional space ($K \ll N$), and preserving the local structural features of the image without loss of information.

A. STACKED DENOISING AUTO-ENCODER

Auto-encoder is an unsupervised dimensionality reduction learning algorithm, which is similar to principal component analysis (PCA). But it is more powerful than PCA, because its hidden layer acts as a feature extractor. In order to learn more robust features, the risk of the constant function is solved by randomly using the damaged input to the network. Denoising auto-encoder (DAE) [41] forcing the auto-encoder to remove the noise and discover more robust features to reconstruct the input. Based on the multi-layer perceptron (MLP) architecture, multiple single-layer denoising auto-encoders are stacked to map a set of input vectors to a set of output vectors to form a stacked denoising auto-encoder network (SDAE). SDAE networks usually contain three layer structures: i) Input layer x; ii) Hidden layer h; iii) Output layer y. The input layer accepts the raw data with noise added as:

$$\begin{cases} \left\{x_n^m\right\}_{n=1}^N \xrightarrow{\text{Add Noise}} \left\{\tilde{x}_n^m\right\}_{n=1}^N \\ \tilde{x}^m = x^m + \varepsilon^m \\ \varepsilon^m \sim E\left(\vartheta\right) \end{cases} \tag{1}$$

where n and m represent the number and dimensions of the original data, $E(\vartheta)$ is the distribution type of noise, and ϑ is the distribution parameter. Each hidden layer contains a number of fully connected nodes, each of which computes a smooth nonlinear activation function from the input data. Usually, a sigmoid function that maps data to a range between (0, 1), the output of the function is set to h_i:

$$\begin{aligned} h_i &= f\left(\tilde{x}\right) = sigmoid\left(W_i\tilde{x} + b_i\right) \\ &= \frac{1}{1 + exp\left(-\sum_{n=1}^{N} W_n\tilde{x}_n - b_n\right)} \end{aligned} \tag{2}$$

The output of the previous hidden layer is used as the input of the next hidden layer, then map back and reconstruct the original input y_j:

$$y_j = g\left(h\right) = sigmoid\left(W_j^{'}h + b_j^{'}\right) \tag{3}$$

The y is considered to be the prediction of the original input x when given coding h. The weight matrix W of the inverse mapping may be selected as a transposed matrix of the forward mapping: $W' = W^T$, was called tied weights. It can reduce parameters in the network and speed up training.

In addition, different optimization objective functions can be constructed according to different loss criteria (such as energy, entropy, etc.). The following optimization objective function based on cross entropy loss is:

$$J = KL\left(\tilde{x}, y\right) = \sum_{n=1}^{N} \tilde{x}_n \log^{\left(\frac{y_n}{\tilde{x}_n}\right)} + \left(1 - \tilde{x}_n\right) \log^{\left(\frac{1-\tilde{x}_n}{1-y_n}\right)} \tag{4}$$

Usually, the optimization objective function is a convex optimization problem, and the auto-encoder uses a stochastic gradient descent (SGD) iterative algorithm to solve the problem [42]. After several epochs, the loss function converges to a local minimum and the algorithm stops. We are interested in the feature information learned from the hidden layer during the network training process, rather than the reconstruction information of the output layer. The weight W and bias b of each hidden layer are used as feature extractors.

B. GRAPH REGULARIZATION AUTO-ENCODER

In machine learning, usually, the distance between the data points and the mapping function are defined in the European space. There are many successful manifold learning algorithms that conform to the locally invariant characteristics [43], [44]. However, these methods are linear, in practice, some nonlinear data may not be distributed in the European space. So it does not provide enough expressive power to find the representation space that can preserve the local geometry, which requires the introduction of new assumptions about the distribution of data.

The combination of graph regularizer and auto-encoder can achieve nonlinear dimensionality reduction and feature extraction. Laplacian regularized auto-encoder (LAE) [45] formalizes the loss function equation:

$$Cost = \Delta\left(\tilde{x}, y\right) + \lambda\phi\left(H\right) \tag{5}$$

The $\Delta\left(\tilde{x}, y\right)$ is the loss term of the auto-encoder, $\lambda\phi\left(H\right)$ is the item of the graph regularizer, and λ is the penalty coefficient of the training graph regularizer. The learned local

geometry $\phi(H)$ can be expressed as the following:

$$\begin{aligned}\phi(H) &= \frac{1}{2}\sum_i\sum_j S_{ij}\left\|h_i - h_j\right\|^2 \\ &= \frac{1}{2}\left(\sum_i h_i^T \sum_j S_{ij}h_i + \sum_j h_j^T \sum_i S_{ij}h_j \right. \\ &\quad \left. - 2\sum_i\sum_j h_i^T S_{ij}h_j\right) \\ &= \frac{1}{2}\left(tr\left(H^T D_1 H\right) + tr\left(H^T D_2 H\right) - 2tr\left(H^T SH\right)\right) \\ &= tr\left(H^T LH\right) \end{aligned} \tag{6}$$

where S_{ij} is the similarity between data samples x_i and x_j, S is the entire weight matrix. Given n points to form a point set $N = \{x_1, x_2 \ldots, x_n\}$, then use K-Nearest Neighbor (K-NN) algorithm to construct a set of K-NNs for each sample point in N. If x_i is in the K-NN set of x_j or x_j is in the K-NN set of x_i (this relationship is symmetric), x_i and x_j form an edge. Then the weight S_{ij} is set to the weighted distance between the two data points, usually weighted by the method of Heat Kernel on the basis of the Euclidean distance, otherwise S_{ij} is set to zero:

$$S_{ij} = \begin{cases} \exp^{\left(-\|x_i - x_j\|^2 / t\right)}, & t \in R \\ 0, \end{cases} \tag{7}$$

We have formed an edge set $E = \left\{e\left(x_i, x_j\right)\right\}$, $i, j \in n$ based on the point set N and the K-NN algorithm, and a weighting map $G = (N, E)$ is established. Based on the local invariance assumption, we consider mapping the weighted graph to another k-dimensional European space $H = \{h_1, h_2, ..., h_k\}, k \ll n$, make the points in this space maintain the same local geometry as the original data. Where h_i and h_j are the close mappings corresponding to the original data in the new space. By minimizing $\phi(H)$, x_i and x_j are close to ensure that h_i and h_j are also close to each other. The $D_1 = D_2$ are diagonal weight matrices,

$$D_1 = D_{ii} = \sum_j S_{ij}, \quad D_2 = D_{jj} = \sum_i S_{ij},$$
$$L = \frac{(D_1 + D_2 - 2S)}{2},$$

The L is called a Laplacian matrix, and $tr(\cdot)$ is a trace of a matrix.

If implemented in the form of an auto-encoder, f denotes the coding part of the similarity weight matrix S, and g denotes the decoding part. The MLP-based graph regularization structure can provides a more smooth way of enforcing the local geometric structure. The layer-by-layer pre-training structure enables the graph regularization structure of each layer to complete the learning process of a geometric feature, which is usually better than the single-layer graph regularization structure. The similarity weight matrix $S \in R^{n\times n}$ is regarded as n regular n-dimensional original input samples, used for layer-by-layer training, thus:

$$\begin{aligned}\arg\min_{HDH^T=I} tr\left(H^T LH\right) &\rightarrow \arg\min_{rank(H)=k} \left\|S - HH^T\right\|_F^2 \\ &\rightarrow \arg\min_{f,g} \|S - g(f(S))\|_F^2 \\ &\rightarrow \arg\min_{f,g} \|S - g(H)\|_F^2 \end{aligned} \tag{8}$$

The constraint $HDH^T = I$ removes any scaling factor in the embedded space. Based on formula $L = D - S$ and Eckart-Young-Mirsky theorem (ie $H \in R^{n\times k}$ contains k eigenvectors that provide reconstruction of the best rank k of rank S under the Frobenuis norm), we can get $\arg\min_{rank(H)=k} \left\|S - HH^T\right\|_F^2$.

Our G-SDAE network divides joint loss function into independent SDAE loss term and g-regularizer loss term. In (9), which avoids the process of extracting features from being disturbed, resulting in features being impure:

$$Cost_{Joint} = \underbrace{\Delta(\tilde{x}, y)}_{SDAE} + \underbrace{\gamma\Delta(S, g(H))}_{G-regularizer} \tag{9}$$

In addition, we abandoned the design of the simplified anchor point graph [39]. Because it increases the complexity of the network, and the parameters such as the number of anchor points need to be repeatedly adjusted. The anchor improper selection of points may lose many important local geometric relationships in the original image.

III. OVERVIEW OF DETECTING LOOP CLOSURE

The G-SDAE network training part is mainly divided into three parts: i) As shown in Fig. 1(a), the SDAE network pre-training the image data, and selects a part of the training sets as a validation set. Then fine-tuning the hyper-parameter such as the number of nodes, the number of iterations and the learning rate to optimization network model; ii) As shown in Fig.1(b), the graph regularization auto-encoder pre-training and fine-tuning the similarity weight matrix S data, and reconstructs the matrix S to learn the local geometry structure from the original data; iii) As shown in Fig. 1(c), the graph regularization auto-encoder shares the coding part of the SDAE network for joint fine-tuning (the coding part of the SDAE network as the coding part of the G-SDAE network) to obtain an optimal solution.

Hidden layer weights W and biases b in a well-trained network model constitute a feature extractor as (2). The schematic diagram of the whole process of detect the loop closure relationship is shown in Fig. 2. Firstly, selectively extract some interesting or noteworthy parts of the image using sparse key point detection algorithms such as SIFT [21], SURF [22] or ORB [27] feature detection. Second, we sort the detected points in descending order according to the order of key point responses. The first N key points are selected, and the original key-frames gray data is segmented into N image patches of dimension $s \times s$ centered on their

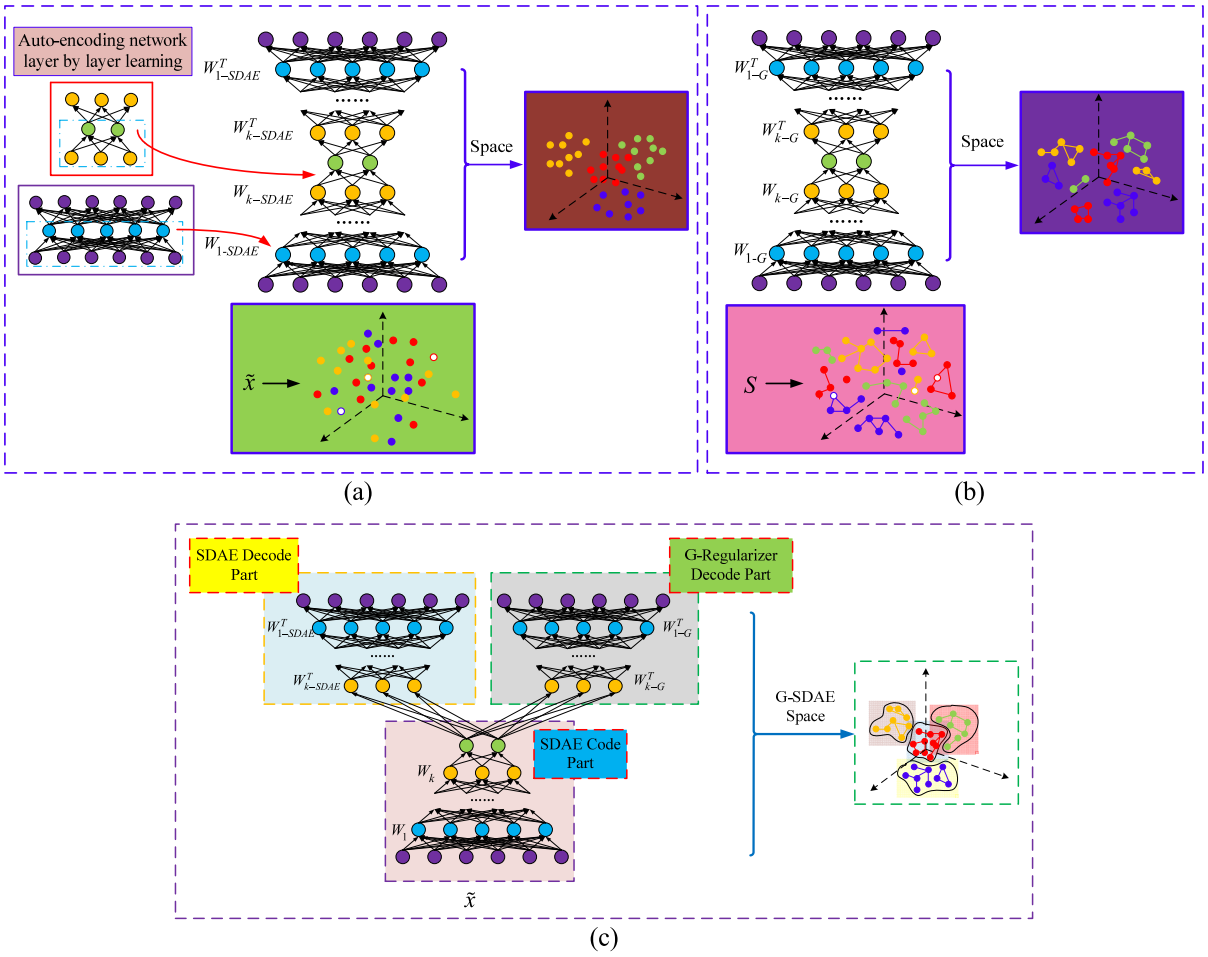

FIGURE 1. G-SDAE network architecture construction. (a) Independent pre-training and fine-tuning of SDAE. (b) Independent pre-training and fine-tuning of graph regularization auto-encoder. (c) G-SDAE network joint fine-tuning.

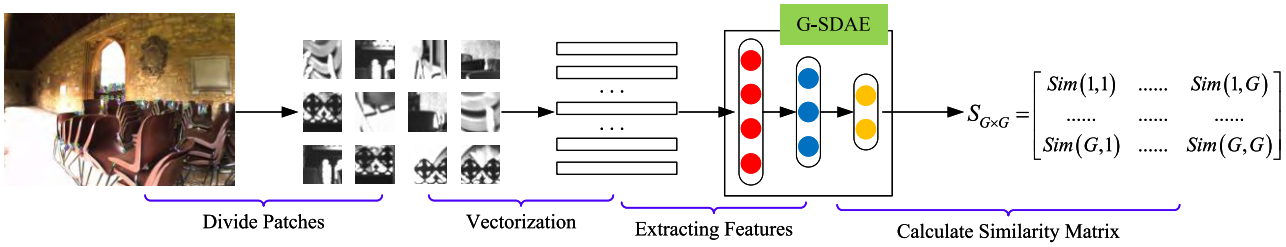

FIGURE 2. Overview of the detecting loop closure process. From left to right: The original grayscale image is segmented according to the key points; raw data vectorization; vectorized data is sent to the well-trained G-SDAE network for feature extraction; calculating the similarity matrix according to the feature and detect the loop closure.

coordinates, and each image patch is pull into a $1 \times s^2$ row vector. The original image is processed into an input matrix $X_{N\times s^2}$, which passes through the feature extractor of each hidden layer in turn until the hidden layer H_f whose final dimension is f. The final hidden layer is used as the feature output layer with f-dimension, and the final output of the original data has a feature matrix of $F_{N\times f}$. Finally, the similarity matching is performed according to the feature matrix, and calculate the similarity scores between each other to determine the loop closure relationship.

A. CALCULATE IMAGE SIMILARITY

Each key-frame $F^{(i)}$ is divided into k high response image patches by the traditional feature extraction algorithm. The well-trained G-SDAE network hidden layer feature extractors obtain features h_k, including abstract features and local geometric structures, and all the features obtained are:

$$F^{(i)} = \{h_1^{(i)}, h_2^{(i)}, \ldots, h_k^{(i)}\}.$$

We use the FLANN (Fast Library for Approximate Nearest Neighbors) [46] algorithm in OpenCV to match these abstract

TABLE 1. Properties of the datasets used.

Dataset	Environment	Conditions	Camera position	Image size	Ground truth
Fr3_ Office	Indoors	Static	Frontal	640 × 480	No
Fr3_Texture	Indoors	Static	Frontal	640 × 480	No
New College	Outdoor	Dynamic	Left and Right	640 × 480	Yes
City Centre	Outdoors	Dynamic	Left and Right	640 × 480	Yes

feature matrices. Each key-frame obtain k feature match-pairs, and then calculating their standard Euclidean distance. The standard Euclidean distance overcomes the shortcoming of different distributions of the various dimensions in the Euclidean distance, so that each dimension satisfies the standard normal distribution:

$$Seuclidean_n^m = \sqrt{\sum_{f=1}^{N_f}\left(\frac{M_n^f - M_m^f}{\delta_f}\right)^2}, \quad n < k,\ m < k \tag{10}$$

The N_f is the dimension of the hidden layer, M is the abstract feature matrix of the hidden layer output, and δ_f is the variance of the f-th dimension. In order to measure the similarity of the abstract feature matrix in the direction, the cosine distance is introduced:

$$Cosine_n^m = \frac{\left(\overrightarrow{M_n} \times \overrightarrow{M_m}\right)}{(\|M_n\| \times \|M_m\|)}, \quad n < k,\ m < k \tag{11}$$

The Euclidean distance measures the distance of points on two high dimensional spaces, and the cosine distance pays more attention to the difference in direction between the two vectors. The detailed process of calculating the similarity is shown in Algorithm 1:

Algorithm 1 Calculate Image Distance

1: Input: Two abstract feature matrix M_1, M_2
▷ Get matching pairs using OpenCV tools
2:*Matching_pair* ← *FLANN_function*(M_1, M_2)
▷ Loop over feature matrix matching pair
3: **for** *i*, *j in range*(*Matching_pair*)**do**
▷ Calculate standard Euclidean distance of the matched patch
4: *Euclidean* ← *pdist*(($M_1[i], M_2[j]$),′ *seuclidean*′)
▷ Calculate the cosine distance of the matched patch
5: *Cosine* ← *pdist*(($M_1[i], M_2[j]$),′ *cosine*′)
▷ Calculate the total distance
6: *Dis_sum* ← *Sum*(*Euclidean*, *Cosine*)
▷ Superimpose the distance of matching pairs
7: *Dis_frame* ← *Sum*(*Dis_frame*, *Dis_sum*)
8:**end for**
9: Output: The distance of feature matrices *Dis_frame*

In addition, it is also necessary to convert the inverse relationship between distance and similarity into a proportional relationship. We count the minimum distance *Min_dis* and the maximum distance *Max_dis* between the abstract feature matrix. Then, using the normalization method (Min-Max) to convert the distance to a similarity score with a positive correlation and quantized to [0, 1]:

$$Sim_j^i = 1 - \frac{Dis\_frame_j^i - Min\_dis_i}{Max\_dis_i - Min\_dis_i}, \quad i < k,\ j < k \tag{12}$$

IV. EXPERIMENT AND RESULT

Experimental validation of our approach has been performed on four publicly available datasets containing dynamic/static image sequences namely ''City Centre'', ''New College'', ''Fr3_Office'' and ''Fr3_Texture'' [14], [47], [48]. We select the mainstream BoW method, the OpenFABMAP detection algorithm, the traditional SDAE network and G-SDAE network to test the above datasets, respectively. The OpenFABMAP is an improved algorithm based on the FAB-MAP (Fast Appearance-Based Mapping) algorithm proposed by Glover [49]. Our experiments are based on the stacked denoising auto-encoder reference code provided by Theano (open deep learning framework) and DELL OPTIPLEX 9020 (CPU: 3.60GHz, 4 Cores. Memory: 12GiB) [50].

The ''Fr3_Office'' dataset uses ASUS Xtion sensors to collect image data along a desk and there is a large loop closure path. The ''Fr3_Texture'' dataset consists of several conference posters and the trajectories overlap at the beginning and end. The ''New College'' and ''City Centre'' are often used in visual SLAM loop closure detection studies. The left and right side cameras of the mobile robot collect images approximately every 1.5 meters. The datasets are mainly from urban outdoor environments under different dynamic conditions, including some chairs, pedestrians, trees, cars, etc. The relevant information of the datasets is shown in Table 1.

A. DATA PREPROCESSING

In practical applications, because of the limited computing power of the mobile platform, we cannot handle all images collected during the motion. It is of little significance to verify the similarities between images with short acquisition intervals, we need to clean the datasets. Firstly, the ORB [27] is used to extract features and descriptors, and the PnP (Perspective-n-Point) method is used to solve the motion between position point pairs. When the motion estimation between the two frames is correct, the feature matching is successful and the motion distance is within the appropriate

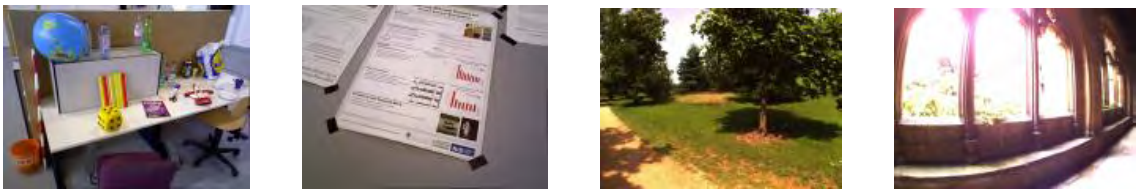

FIGURE 3. Sample image randomly selected from the datasets.

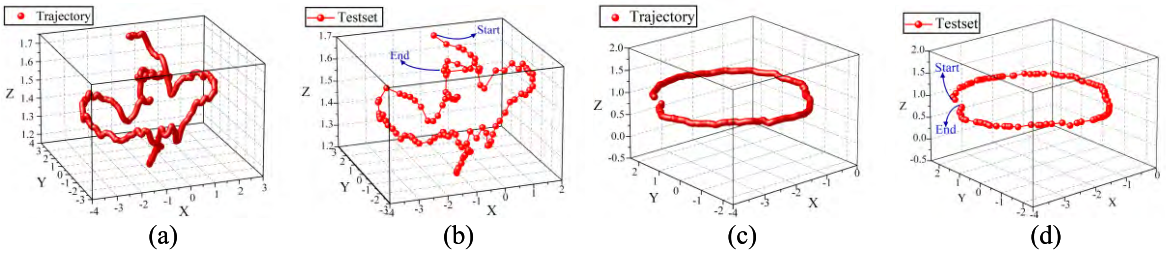

FIGURE 4. Original trajectory and testsets key-frames. (a) The original trajectory of the "Fr3_Office" dataset. (b) Keyframes in the "Fr3_Office" testsets. (c) The original trajectory of the "Fr3_Texture" dataset. (d) Key-frames in the "Fr3_Texture" testsets.

TABLE 2. Training sets and testsets.

Dataset	Original data	Training sets	Testsets
Fr3_ Office	2585	267	107
Fr3_Texture	1634	234	104
New College_Left	1073	495	248
New College_Right	1073	495	248
City Centre_Left	1237	619	309
City Centre_Right	1237	619	309

interval, this frame is used as the training data. The number of training sets and test sets we get is displayed on Table 2. In addition, Fig. 3 shows some of the filtered data samples and Fig. 4 shows the 3D trajectory of the original datasets and the filtered test sets (''Fr3_Office'' and ''Fr3_Texture'').

B. GROUND TRUTH

Although some datasets have a loop closure path, but the official does not provide ground truth, we need to design the algorithm to provide ground truth with true loop closure relationship. The algorithm flow for obtain the ground truth is shown in Fig. 5. We first extract the ORB features from the RGB images provided by the datasets and make a match. When the matching number of descriptor pair and the descriptor matching distance are within a certain threshold interval. Then, we use the SVD method to solve the ICP (Iterative Closest Point) for the depth data that is to solve the motion estimation problem between the two matching points. We optimize the obtained rotation matrix R and translation vector t through the pose graph tools $g2o$ [5].

Finally, the rotation matrix is $R_{3\times3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, and convert it into Euler angles corresponding to three coordinate axes:

$$\begin{aligned} \theta_x &= atan2(r_{32}, r_{33}) \\ \theta_y &= atan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\ \theta_z &= atan2(r_{21}, r_{11}) \end{aligned} \tag{13}$$

We add the translation vector $t_{1\times3}$ to each dimension of the corresponding Euler angle $(\theta_x, \theta_y, \theta_z)$, respectively, and convert them into similarity scores sim for judging whether two images are loop closure. The resulting ground truth information has visualized in Fig. 8 (a) and Fig. 8(c).

TABLE 3. Parameters needed to train the network.

Parameter	Symbol	Default value
Pre-training rate	h	0.1
Fine-tuning rate	z	0.05
Joint fine-tuning rate	m	0.01
Corruption	c	0.2
Batch size	b	60
Size of image patch	S_l	41×41
Pre-training epochs	N_t	80
Fine-tuning epochs	Nt	50
Hidden layer structure	H	2000-1500-1000-500

C. PERFORMANCE ANALYSIS

For the BoW method, we use the open source bag-of-word model DBoW3 to generate the ORB feature dictionary. We use the default parameters in the OpenFABMAP algorithm provided by the author. Based on experience, the more hidden layers of the auto-encoder, the more abstract the extracted feature information is. Therefore, the SDAE and

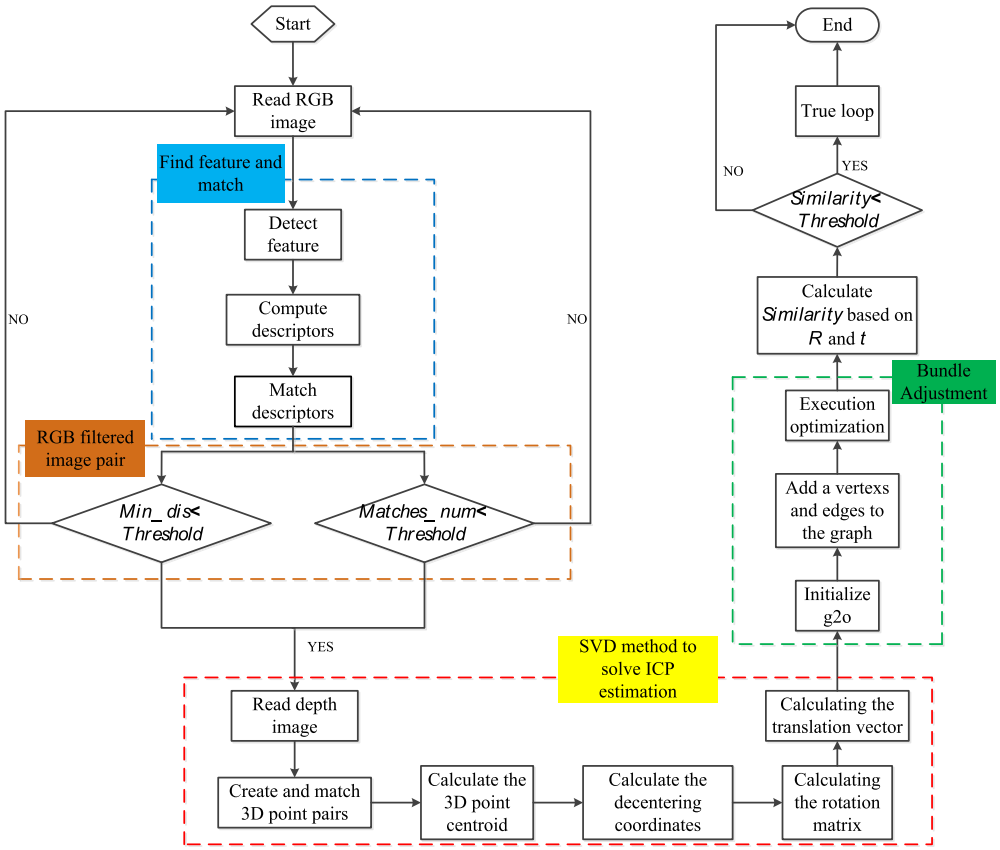

FIGURE 5. Algorithm flow for obtaining ground truth.

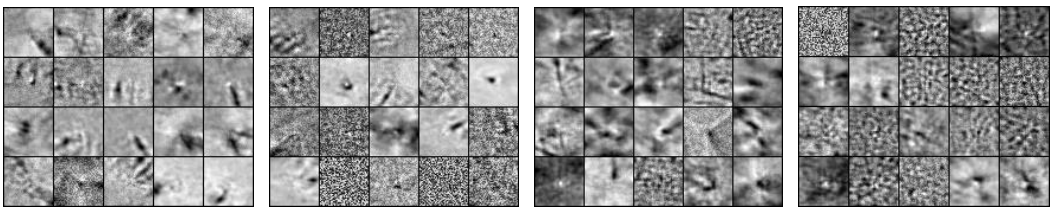

FIGURE 6. Hidden layer feature visualization. (a) Features extracted from "Fr3_Office" (G-SDAE). (b) Features extracted from "Fr3_Office" (SDAE). (c) Features extracted from "Fr3_Texture" (G-SDAE). (d) Features extracted from "Fr3_Texture" (SDAE).

G-SDAE networks all adopt a four layers network structure. The default training parameters of the network are shown in Table 3.

1) INDOOR STATIC SCENE

In an auto-encoder, the weight matrix W of the first hidden layer is treated as a feature extractor. We calculate the dot product of the input data and the weight matrix W, when the input of the first hidden layer is x, the number of rows of the matrix W is the same as the number of columns of x. The features acquired by the weight W of the first hidden layer can be visualized. Partial visualization features extracted from the "Fr3_Office" and "Fr3_Texture" by the first hidden layer of the SDAE network and the G-SDAE network are shown in Fig. 6, these features are collected in the same position of the weight matrix W.

The first layer of hidden layer learn some meaningful features, including some prominent corner points and distinct edges. It can be seen from Fig. 6 that many of the hidden units in the hidden layer of the SDAE network have a low average response, only part of them are detecting useful information. However, there are more hidden layer units activated in the G-SDAE network (the area shown as a prominent color in the figure). Those nodes that are not activated cannot learn useful information, and represented in the form of random noise in the figure, which is related to the initialization of

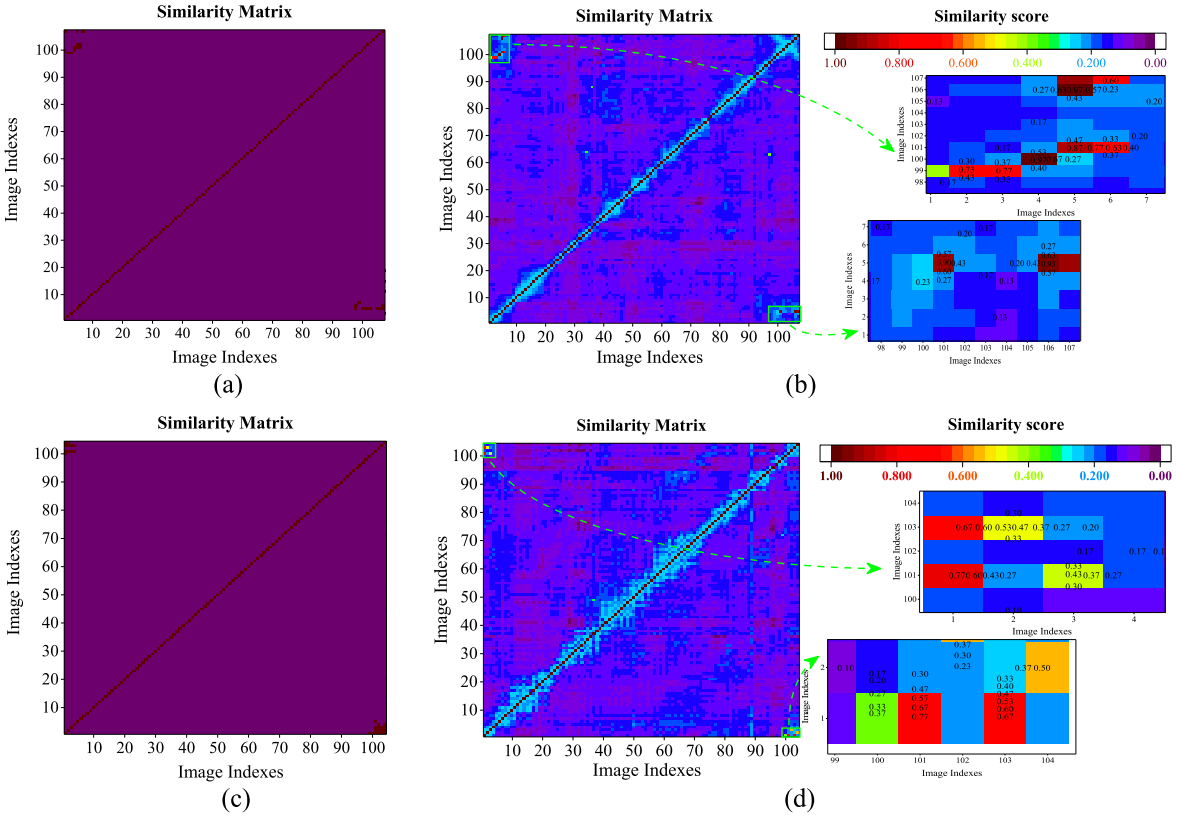

FIGURE 7. The similarity score matrix. (a) The similarity score matrix obtained by ground truth ("Fr3_Office"). (b) The similarity score matrix obtained by G-SDAE ("Fr3_Office"). (c) The similarity score matrix obtained by ground truth ("Fr3_Texture"). (d) The similarity score matrix obtained by G-SDAE ("Fr3_Texture").

network weights. The response of hidden units forms a nonlinear description of the image data. Their output is regarded as features which are used for measuring the similarity of the input data.

As shown in Fig. 7(b) and Fig. 7(d), we match the key-frames similarity scores and form a normalized similarity matrix by (13) and Algorithm 1. We can more intuitive to see the loop closure relationship between the images. In addition, the similarity score matrix of the ground truth is shown in Fig. 7(a) and Fig. 7(c).

Comparing the ground truth with the similarity score matrix obtained by training the G-SDAE network, the loop closure information is concentrated in the upper left corner and the lower right corner. It represents some key-frames at the beginning of the motion and some key-frames at the end of the motion, which is consistent with the fact that there are only one large loop closure in the two static datasets. As shown in Fig.7(b) and Fig. 7(d), the similarity score between some images at the beginning and end of the motion trajectory is higher than 0.5, and they are regarded as true loop closure. For other key-frames that do not have a loop closure relationship, such as those key-frames that are close to the current key-frame, the similarity score is reasonably assigned a smaller score. As shown in Fig. 8, we show the detected loop closure match pairs. Fig. 8(a) and Fig. 8(c) mark the true loop closure relationship in ground truth, respectively. Fig. 8(b) and Fig. 8(d) mark the loop closure relationship matched by the G-SDAE network, respectively. The key-frames of the loop closure are connected by a blue 3D ball and a cyan 3D ball through a yellow arrow.

In order to compare the results of various loop closure detection methods, we scan the similarity score threshold to obtain a P-R curve. The P-R curve represents the relationship between precision rate and recall rate. Precision rate is defined as the ratio between accurately detected loop closure frames (True-Positive) and the total number of detections returned by the method (True-Positive plus False-Positive). Additionally, recall rate is defined as the number of True-Positive detections found, over the total number of loop closure frames that exist in the ground truth (True-Positive plus False-Negative). As shown in Fig. 9(a) and Fig. 9(b), we compared the similarity scores from four different methods, including the DBoW3 bag-of-words model, the OpenFABMAP algorithm, the SDAE network and the G-SDAE network with different hidden layers.

In SLAM, the recall rate is relatively loose, and more attention is paid to the precision rate, because the false positive loop closure will add the wrong edge to the backend Pose Graph, and the optimization algorithm may give a completely

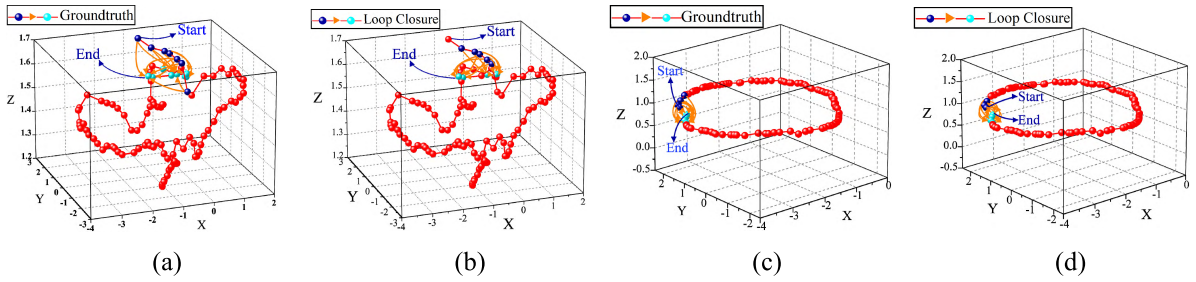

FIGURE 8. Loop closure matching key-frames. (a) Loop closure detected by ground truth ("Fr3_Office"). (b) Loop closure detected by G-SDAE ("Fr3_Office"). (c) Loop closure detected by ground truth ("Fr3_Texture"). (d) Loop closure detected by G-SDAE ("Fr3_Texture").

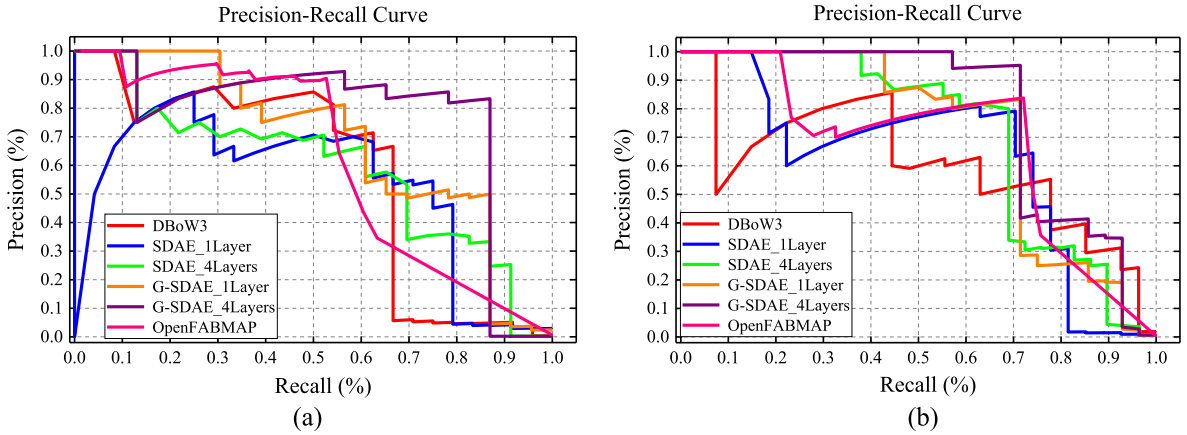

FIGURE 9. P-R curve obtained by using different methods. (a) P-R curves obtained from the "Fr3_Office" dataset. (b) P-R curves obtained from the "Fr3_Texture" dataset.

wrong result. It may lead to the curvature of the corridors in the established map, the staggering of the walls, and the failure of the entire map. In contrast, the recall rate is lower, some true loop closures are not detected, and the constructed map is affected by certain errors, but it can be repaired after several loop closure detection processes. According to the results shown in Fig. 9(a), when the recall rate of the G-SDAE network is greater than 85%, the precision rate can still be maintained above 80%. In other methods, when the recall rate is within the range of (70%, 85%), the precision rate is below 50%. When the recall rate is less than 60%, the threshold of the detecting rises at this time, and fewer loop closure are detected, the precision rate of each method is higher than 60%. As shown in Fig. 9(b), when the recall rate is below 70%, the precision rate of the G-SDAE_1Layer method and the SDAE_4Layer method are both greater than 80%, and the precision rate of the G-SDAE_4Layer method is greater than 90%. By analyzing the impact of the hidden layers of the network on the experimental results, we find that the traditional SDAE network and the G-SDAE network follow the progressive relationship between the hidden layers. In other words, a network with a deep hidden layer structure is more efficient than a network with a shallow hidden layer structure.

2) OUTDOOR DYNAMIC SCENE

Early place recognition systems often implicitly used the simplifying assumption that the visual appearance of each place would not change over the course of the experiment. However, it is clear that the appearance of a place can vary greatly over time due to a large number of causes including changes in lighting and weather. This will lead to an increase in the false positive rate, which increase the difficulty of accurately detecting the loop closure. We divided the "New College" and "City Centre" into two experimental datasets, separately, the left side view and the right side view. In addition, the similarity matrix heat map is shown in Fig. 10, and each set of figures in Fig. 10 contains the heat map drawn by the ground truth (on the left side) and the heat map obtained by the G-SDAE network (on the right side).

Both public datasets also provide aerial photos of the data collection area to visualize the results. According to the officially provided GPS positioning data of the robot movement, we can draw a moving trajectory based on the real scene. The similarity score within a certain threshold range obtained by the G-SDAE network is considered as key-frames pairs with true loop closure relationship. For example, based on the trajectory of the real scene, we show the loop closure key-frames detected in the data collected by the left

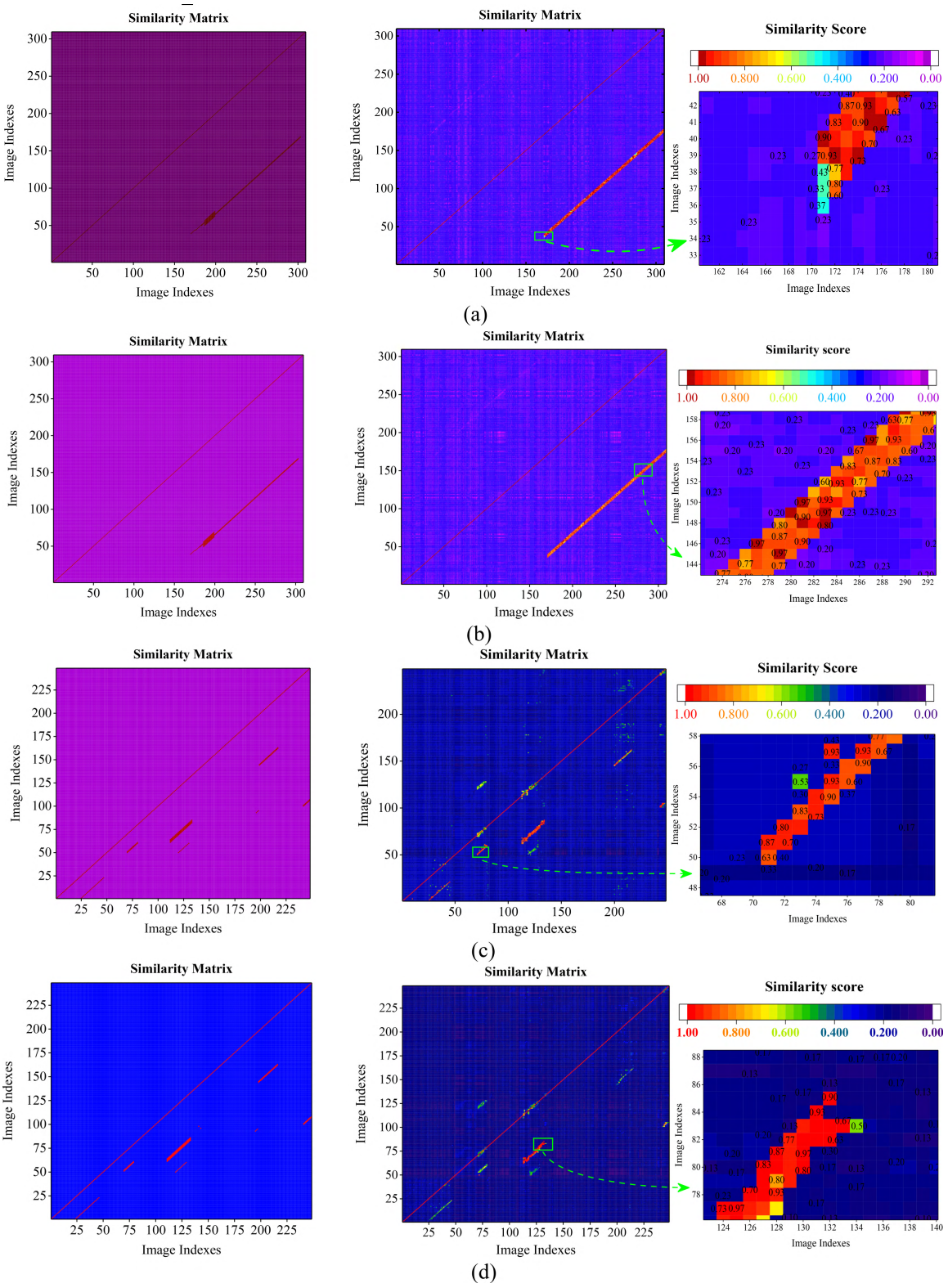

FIGURE 10. The similarity score matrix. (a) "City Centre_Left". (b) "City Centre_Right". (c) "New College_Left". (d) "New College_Right".

camera of "New College" and "City Centre" in Fig. 11, Fig. 12 and Fig. 13.

The loop closure key-frames are marked in red and are connected by solid green lines. The Fig. 11 shows the

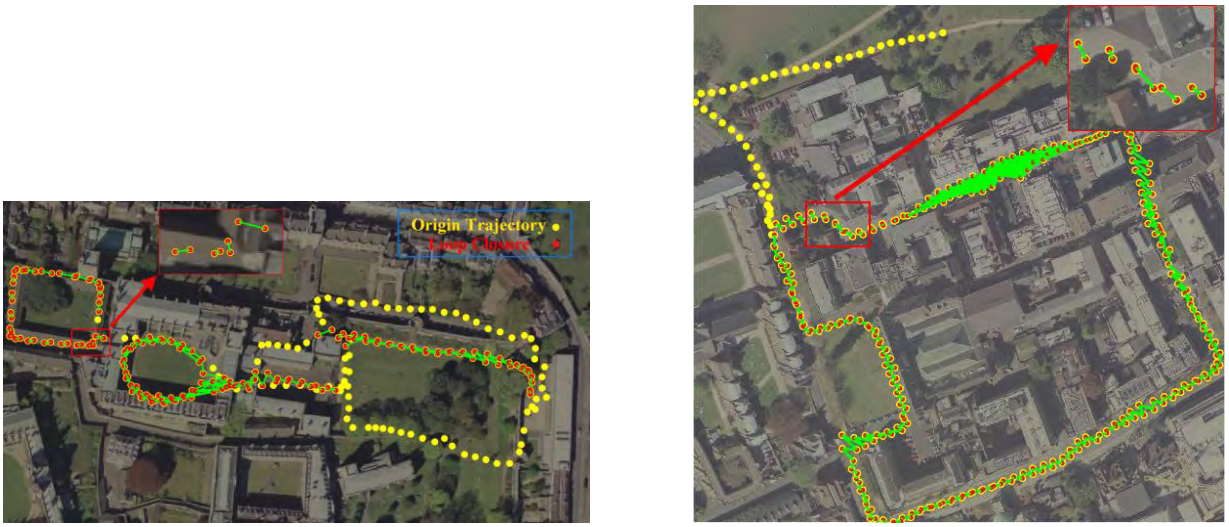

FIGURE 11. Motion trajectory and loop closure obtained by ground truth (New college is on the left and city center is on the right).

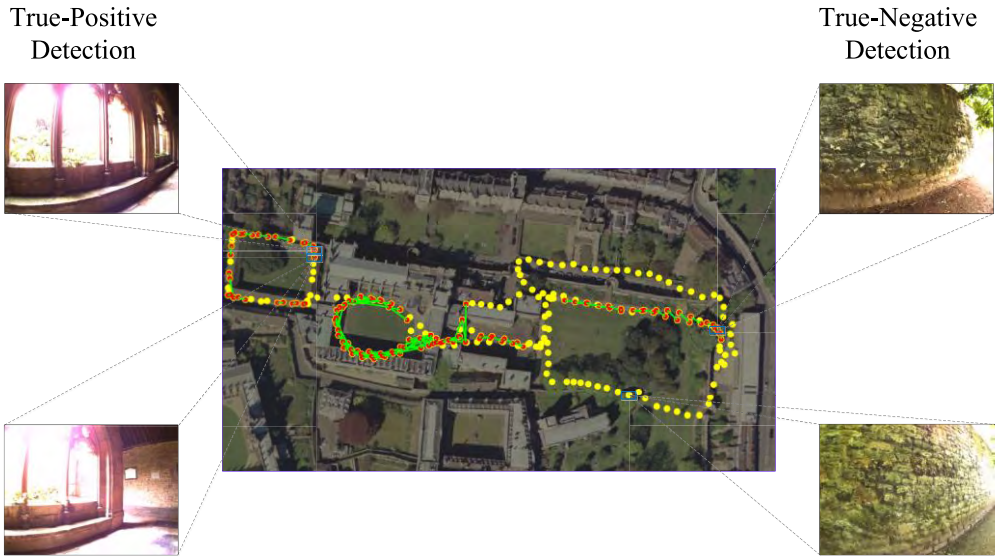

FIGURE 12. Loop closure detection results on the "New College_Left" dataset. representative true-negative and true-positive examples are highlighted.

loop closure detected by ground truth, the Fig. 12 and Fig. 13 shows the loop closure detected by the G-SDAE network. Similarly, based on these two outdoor dynamic datasets, we plot the corresponding P-R curves in Fig. 14. It can be seen from Fig. 14 that the P-R curve obtained by the G-SDAE network has the highest precision rate in the same recall rate, and the DBoW3 method is the worst.

3) AVERAGE-PRECISION (A-P) SCORE COMPARISON

Sometimes it is necessary to use the A-P score to balance the precision rate and recall rate. The A-P score refers to the area under the P-R curve, and the A-P score represents the average of the precision rate on the different recall rate. It represents a weighted average of the precision rate achieved at each similarity score threshold, and using the increase in the recall rate of the previous threshold as the weight:

$$AP = \sum_{n} (R_n - R_{N-1}) P_n \tag{14}$$

where P_n and R_n are the precision rate and recall rate at the nth threshold. This implementation is not interpolated, unlike using the trapezoidal rule to calculate the area under the P-R curve. The trapezoidal rule uses linear interpolation and may be too optimistic. The A-P scores obtained by the above all methods for all public datasets involved in the experiment are shown in Table 4 and Fig. 15.

From Table 4, the calculated A-P scores can be more clearly seen that the G-SDAE network has the highest A-P score in all datasets. By appropriately increasing the depth

FIGURE 13. Loop closure detection results on the "City Centre_Left" dataset. representative true-negative and true-positive examples are highlighted.

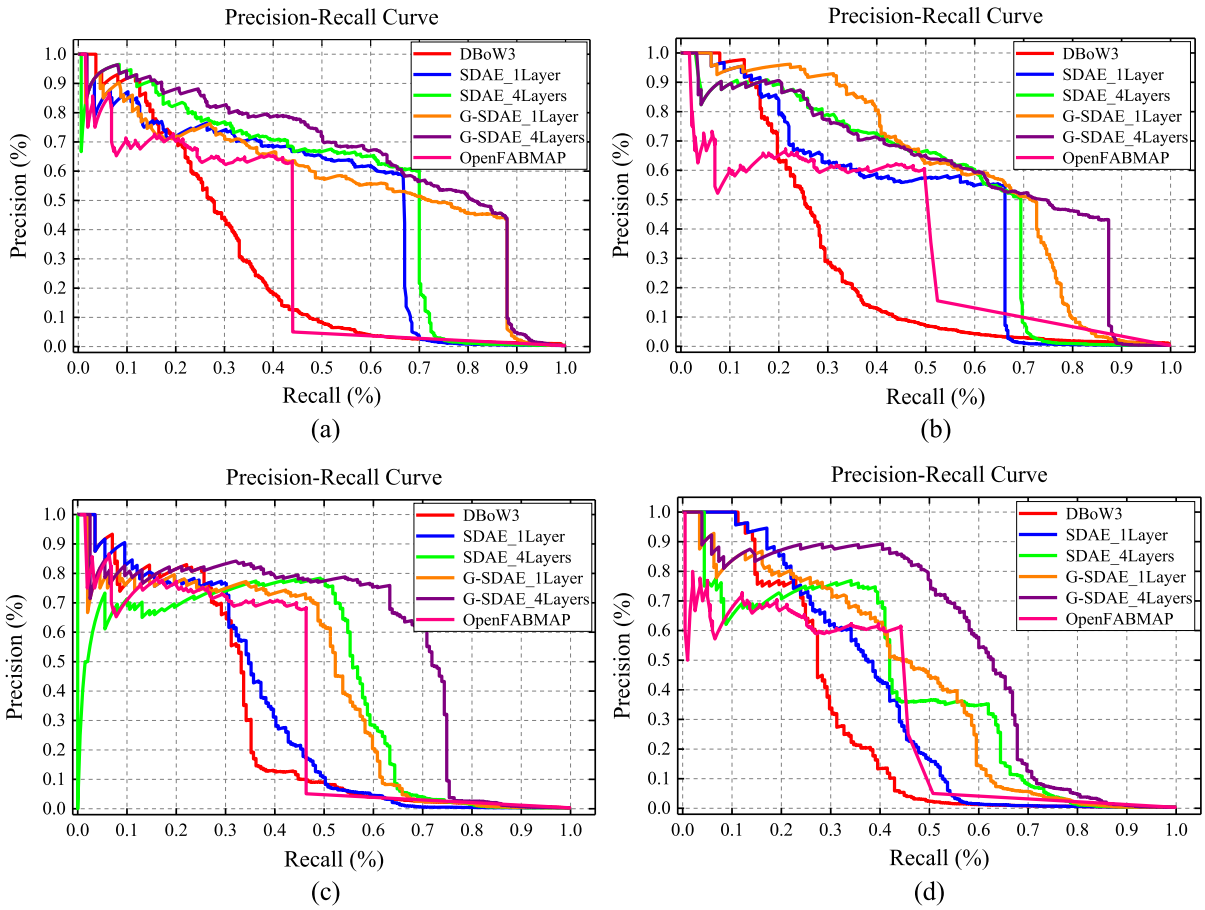

FIGURE 14. P-R curves obtained using different methods in the "City Centre" and "New College" datasets. (a) P-R curve drawn by the "City Centre_Left" dataset. (b) P-R curve drawn by the "City Centre_Right" dataset. (c) P-R curve drawn by the "New College_Left" dataset. (d) P-R curve drawn by the "New College_Right" dataset.

TABLE 4. Average-precision (A-P) score calculated for various experimental methods on different datasets.

Dataset	DBoW3	OpenFABMAP	SDAE_1Layer	SDAE_4Layer	G-SDAE_1Layer	G-SDAE_4Layer
Fr3_Office	0.620574	**0.707956**	0.621042	0.710105	**0.721418**	**0.792804**
Fr3_Texture	0.566410	**0.724671**	0.539159	0.590269	**0.686497**	**0.776243**
City Centre_Left	0.289054	0.482232	0.486958	0.535800	**0.568482**	**0.653369**
City Centre_Right	0.280386	0.468262	0.468091	0.526942	**0.597297**	**0.605748**
New College_Left	0.305506	0.455419	0.328715	0.428794	**0.433665**	**0.581860**
New College_Right	0.282112	**0.525625**	0.358634	0.407247	**0.420058**	**0.555784**

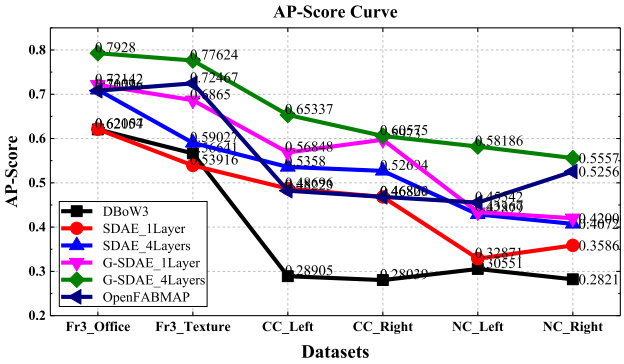

FIGURE 15. Average-precision (A-P) score calculated for various experimental methods on different datasets (CC: City center, NC: New college).

of the network structure, we can get a more distinguishing feature representation, which means that the correct loop closure can be accurately detected.

V. CONCLUSION

This paper focuses on the loop closure detection problem in visual SLAM. In our work, we propose a manifold learning auto-encoder (G-SDAE) based on graph regularizer. The G-SDAE network model learns a powerful and compact embedded space to preserve the geometry structure in a local manifold. According to the experimental results, the average accuracy of the loop closure detection by training the G-SDAE network is higher than that of the DBoW3 method and the traditional SDAE method.

However, some hyper-parameters involved in a deep learning network model usually require repeated experiments to obtain an optimized network structure. The separate pre-training and fine-tuning of the G-SDAE network requires extra time consumption. Therefore, how to simplify the network structure and accelerate the training process, will be the subject of further research.

REFERENCES

[1] R. C. Smith and P. Cheeseman, *On the Representation and Estimation of Spatial Uncertainly*. Thousand Oaks, CA, USA: SAGE, 1986.

[2] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Mach. Intell. Pattern Recognit.*, vol. 5, no. 5, pp. 435–461, 1988.

[3] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[4] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Oct. 2003, pp. 206–211.

[5] R. KÃijmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 3607–3613.

[6] K. L. Ho and P. Newman, "Detecting loop closure with scene sequences," *Int. J. Comput. Vis.*, vol. 74, no. 3, pp. 261–286, Sep. 2007.

[7] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3D modeling and tracking," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 137–147, Jan. 2014.

[8] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.

[9] W. Brian, G. Klein, and I. Reid, "Automatic relocalization and loop closing for real-time monocular SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1699–1712, Sep. 2011.

[10] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular SLAM," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1188–1197, Dec. 2009.

[11] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[12] W. Burgard, O. Brock, and C. Stachniss, "Mapping large loops with a single hand-held camera," in *Robotics: Science and Systems*. Cambridge, MA, USA: MIT Press, 2007, pp. 297–304.

[13] Y. Latif, G. Huang, J. J. Leonard, and J. Neira, "An online sparsity-cognizant loop-closure algorithm for visual navigation," presented at the Robot. Sci. Syst., Berkeley, CA, USA, Jul. 2014.

[14] M. Cummins 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, Jun. 2008.

[15] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Automat.*, Saint Paul, MN, USA, May 2012, pp. 1643–1649.

[16] W. Maddern, M. Milford, and G. Wyeth, "CAT-SLAM: Probabilistic localisation and mapping using a continuous appearance-based trajectory," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 429–451, Apr. 2012.

[17] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with SMART," in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2014, pp. 1612–1618.

[18] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2008, pp. 1828–1833.

[19] M. Labbé and F. Michaud, "Appearance-based loop closure detection for Online large-scale and long-term operation," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 734–745, Jun. 2013.

[20] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1611–1626, Dec. 2013.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[22] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 404–417.

[23] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Prog. Brain Res.*, vol. 155, pp. 23–36, Jan. 2006.

[24] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[25] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Fast loop-closure detection using visual-word-vectors from image sequences," *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 62–82, Jan. 2018.
[26] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2161–2168.
[27] E. Rublee, V, Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[28] G. Singh and J. Kosecka, "Visual loop closing using gist descriptors in manhattan world," presented at the ICRA Omnidirectional Vis. Workshop, Anchorage, AK, USA, Oct. 2010.
[29] Y. Liu and H. Zhang, "Visual loop closure detection with a compact image descriptor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 1051–1056.
[30] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1352–1359.
[31] Y. Xia, J. Li, L. Qi, H. Yu, and J. Dong, "An evaluation of deep learning in loop closure detection for visual SLAM," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data*, Jun. 2017, pp. 85–91.
[32] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[33] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," *Auton. Robots*, vol. 41, no. 1, pp. 1–18, Jan. 2017.
[34] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE CVPR*, Jun. 2014, pp. 806–813.
[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1717–1724.
[36] Y. Hou, H. Zhang, and S. Zhou, "Convolutional neural network-based image representation for visual loop closure detection," in *Proc. IEEE Int. Conf. Inf. Automat.*, Aug. 2015, pp. 2238–2245.
[37] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2015, pp. 4297–4304.
[38] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst*, Nov. 2017, pp. 3856–3866.
[39] S. Yang, L. Li, S. Wang, W. Zhang, and Q. Huang, "A graph regularized deep neural network for unsupervised image representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7053–7061.
[40] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Oct. 2015.
[41] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 1096–1103.
[42] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.
[43] M. Belkin and P. Niyogi. *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*. Cambridge, MA, USA: MIT Press, 2003.
[44] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[45] K. Jia, "Laplacian auto-encoders: An explicit learning of nonlinear data manifold," *Neurocomputing*, vol. 160, pp. 250–260, Jul. 2015.
[46] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, vol. 1, no. 2, pp. 331–340, Feb. 2009.
[47] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *Int. J. Robot. Res.*, vol. 28, no. 5, pp. 595–599, May 2009.
[48] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.
[49] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 4730–4735.
[50] Theano Development Team *et al.* (2016). "Theano: A Python framework for fast computation of mathematical expressions." [Online]. Available: https://arxiv.org/abs/1605.02688

ZHONGHUA WANG was born in Shandong, China, in 1993. He is currently pursuing the master's degree with the School of Information Engineering, Capital Normal University, China. His research interests include visual simultaneous localization and mapping (SLAM) systems, image processing, and deep learning.

ZHEN PENG received the B.S. and M.S. degrees in computer applications technology from Shandong University, Jinan, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer applications technology from the University of Science and Technology Beijing, Beijing, China, in 2011. She is currently an Assistant Professor with the Beijing Institute of Petrochemical Technology. Her research interests include data mining, fault diagnosis, power electronics, and electrical vehicles.

YONG GUAN (M'12) received the Ph.D. degree in information and telecommunication engineer from the China University of Mining and Technology, in 2003. He is currently a Professor with Capital Normal University. His main research interests include formal verification techniques, fault diagnosis, power electronics, and electrical vehicles.

LIFENG WU (M'12) received the B.S. degree in applied physics from the China University of Mining and Technology, in 2002, the M.S. degree in detection technology and automation device from Northeast Electric Power University, in 2005, and the Ph.D. degree in physical electronics from the Beijing University of Posts and Telecommunications, in 2010. From 2012 to 2013, he was a Visiting Scholar with Tsinghua University. From 2014 to 2015, he held a postdoctoral position at the University of Maryland, College Park, MD, USA. From 2017 to 2018, he was a Visiting Scholar with Peking University. He is currently an Assistant Professor with Capital Normal University. His research interests include data-driven modeling, estimation and filtering, fault diagnosis, power electronics, and electrical vehicles.