Local Patch Descriptor Using Deep Convolutional Generative Adversarial Network for Loop Closure Detection in SLAM

Dong-Won Shin and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST),
123 Cheomdangwagi-ro, Buk-gu, Gwangju, 61005, Republic of Korea
E-mail: {dongwonshin, hoyo}@gist.ac.kr

***Abstract*— Recently, the augmented reality and virtual reality fields have been actively researched and a lot of major companies have been aggressively investing the fields. On the core of the research field, the simultaneous localization and mapping (SLAM) algorithm which estimates the camera's position in a global coordinate and simultaneously constructs a 3D environment map firmly settled. Among typical components of modern SLAM framework, we are focusing on a loop closure detection for determining whether the current position of a robot agent was visited previously. The conventional algorithms for the loop closure detection relied on clustering hand-crafted features like SIFT, SURF, and ORB which appear a weakness to handle variations in the image such as a viewpoint change, illumination change, deformation, and occlusion. In this paper, we propose a local patch descriptor using a deep convolutional generative adversarial network to deal with the variations. The experiment result displays the proposed method well clusters the image patches with similar appearances better than the hand-craft features.**

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is an algorithm estimating the camera's position in a global coordinate and simultaneously constructing a 3D environment map surrounding a robot agent. A prediction of the accurate 3D environment map and the camera's position is essential for applications such as augmented reality, virtual reality, and robotics.

First, the augmented reality (AR) is a technique that allows a virtual object to be synthesized in a real world that a user is looking at and gives an immersive feeling as if it exists in front of the user. This is solely possible after achieving the accurate 3D environment map and correct localization of the viewpoint of the user. Second, the virtual reality is a technique bringing a user to a totally virtual world and gives an immersive feeling as if the user is in the space. For this purpose, the camera localization capability is necessary; however, the current VR technologies require external sensory devices. By the SLAM technology with the embedded camera sensor, the cost would become reasonable and the convenience would be increased. Lastly, the robotics will be the most promising field with SLAM in the near future. The perception and understanding of the surrounding environment are the prior part to accomplish sophisticated tasks such as autonomous driving, robot navigation, and drone aviation.

The typical SLAM algorithm is divided into the front-end and the back-end [1]. The front-end performs the process of making the data measured by the sensor into a three-dimensional point cloud and matching process, and the back-end processes the loop closure detection, deformation handling, and optimization of the environment map. In this paper, we investigated the loop closure detection, which is an important part of the overall SLAM algorithm.

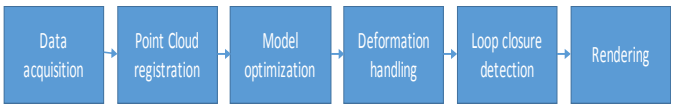

Fig. 1 Typical components of modern SLAM framework

Loop closure detection represents a determining task whether a current position of a robot agent was visited previously, in order to impose a constraint on an environment map optimization in SLAM. This process is necessary to construct a consistent 3D map model and solves a drift problem by correcting the camera trajectories.

Let's assume the triangle is a series of robot trajectories and circles represent the same position as you can see in the Fig. 1. Fig. 2(a) shows that the trajectory of the robot is severely distorted due to the cumulative trajectory error when the loop closure detection is not used. This also directly affects the accuracy of the environmental map. Conversely, when the loop closure detection is used as Fig. 2(b), the trajectory can be properly corrected and an accurate environment map can be obtained at the same time.

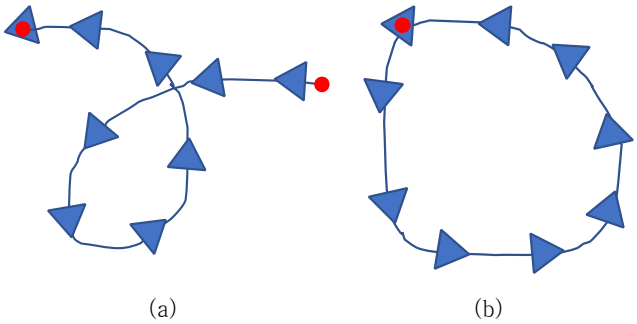

Fig. 2 Example of loop closure detection

II. Related Work

A. Loop Closure Detection

The conventional loop closure detection method is divided into a local image feature-based method and a whole-image descriptor-based method. First, the local image feature method extracts local image features such as SIFT and SURF from the image and then generates an image descriptor based on the feature. The distance between the image descriptors obtained from each image is calculated. If the distance is lower than the predetermined threshold value, it is determined that the current position has been visited before and the loop closure is detected.

A typical method is the bag-of-visual-words method. In this method, the extracted local image features are clustered in a multidimensional feature space. The center of the cluster is defined as a visual word and the collection of visual-words from various images is called as bag-of-visual-words. When an image comes in to acquire the descriptor using the learned bag-of-visual-words, the feature is extracted by the same local image feature detector exploited in the training stage. Then, the vector quantization step is performed and the histogram of the visual-words is calculated as the image descriptor [2].

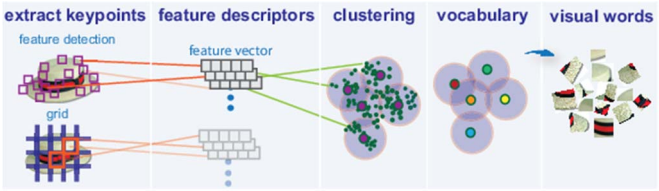

Fig. 3 Bag-of-Visual-Word method

The whole image descriptor-based method is a method of acquiring an image descriptor by utilizing the entire image, rather than analyzing the image as a component such as objects or local features.

Recently, a loop closure detection method based on the whole image descriptor using a convolutional neural network (CNN) method has been proposed [3]. In this method, the ImageNet neural network model for the image classification was employed to detect loop closures by using the feature obtained from each layer as an image descriptor [4]. The structure of the neural network model uses the existing ImageNet, but the training data is learned by using the Places dataset focused on the place recognition. Then, loop closure is detected using the image descriptor extracted from each convolution, pooling, and fully connected layer. In the experiment result of [3], the image descriptor obtained from the last pooling layer has the best loop closure detection result.

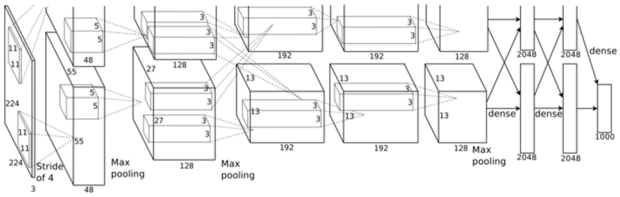

Fig. 4 ImageNet neural network model

B. Generative Adversarial Network

In general, the loop closure detection requires the ability to handle variations such as a viewpoint change, illumination change, deformation, and occlusion in the image since the loop closure doesn't occur at the exact same place, but the slightly deviated place. In order to handle the variations, the descriptor should have a generative capability.

Recently, a lot of generative neural models have been introduced since the astonishing work from Ian Goodfellow. Ian et. al. presented the generative adversarial network (GAN) which produces the competitive relationship between a generator (G) and discriminator (D). The generator generates a fake image imitating a real image and the discriminator discriminates whether the fed input image is real or fake. Along with this competitive training process, the discriminator can represent the images well even in a variety of variations. **Generative** adversarial networkFig. 5 illustrates the generative adversarial network.

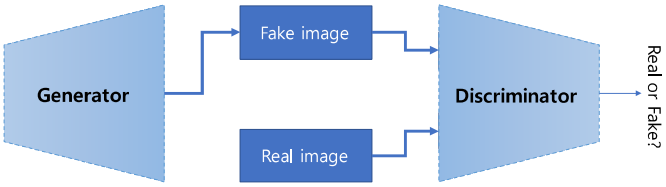

Fig. 5 Generative adversarial network

The competitive relationship of GAN can be mathematically represented by (1). It is a minimax problem of a value function $V(D,G)$. Specifically, the discriminator should maximize this value function which the generator should minimize simultaneously. $D(x)$ outputs a value ranging from zero to one, representing zero for the fake image and one for the real image.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_{x(z)}}[\log(1-D(G(z)))] \quad (1)$$

Some subsequent works based on GAN have been introduced and one of the impressive descendants is the deep convolutional generative adversarial network (DCGAN) which will be used for our experiment. Alec et. al. indicated the architecture guidelines for GAN which makes a huge improvement with following 5 regulations:

- Instead of pooling layers, exploit a strided convolutions and a fractional convolutions.
- Apply batch normalization except for the last layer of the generator and the first layer of the discriminator.
- Remove fully connected layers.
- Use ReLU activation function for all layers in the generator except for the last layer, which uses Tanh.
- Use LeakyReLU activation function for all layers in the discriminator.

By applying the regulations, we can generally achieve higher stability for training the network.

III. PROPOSED METHOD

In this paper, we propose a local patch descriptor using the deep convolutional generative adversarial network. It is a combination between the local feature-based method and the whole image descriptor-based method. Although the proposed method requires more computation time, the accuracy of the clustering capability will be increased.

A. Local Patch Extraction

In order to obtain a training dataset for DCGAN, we first extract the local patches from images by local feature descriptors such as SIFT, SURF, and ORB. We used Places training dataset which contains more than 10 million images comprising 400 unique scene categories, specialized for places recognition tasks. We made the local patches by cropping the image to the size of 64×64 having the local feature point at the center of the image. The number of the local patch per image is restricted to 100 but we removed the local patches near the image boundary which cannot satisfy the size of 64×64. Fig. 6 shows the local patches with SIFT features.

Fig. 6 Local patches with SIFT features

B. DCGAN training

We employed the open source deep learning library, Tensorflow, for our experiment and followed the DCGAN architecture from the literature [5]. The parameters, however, were modified to apply on our dataset as shown in Table 1. For the discriminator, it has four convolutional layers with 5×5 kernel size and stride two. For the generator, it has four fractionally-strided convolution layers with 5×5 kernel size and stride two.

Table 1 Training parameters for our experiment

Parameter	Value	Parameter	Value
Batch size	64	Input height	64
Epoch	25	Input width	64
Learning rate	0.0002	Output height	64
Momentum	0.5	Output width	64

C. DCGAN Descriptor

After training DCGAN model, we can extract descriptors from the discriminator part of the model. The discriminator of DCGAN consists of four convolution layers as shown Fig. 7 and we constructed our local patch descriptor from the output of each layer. Since combining all outputs produces huge feature dimension, we applied the max-pooling operation to reduce the size of the descriptor. The size of the max-pooling operations is different from each layer: (16,16) for layer 1, (8,8) for layer 2, (4,4) for layer 3, and (2,2) for layer 4.

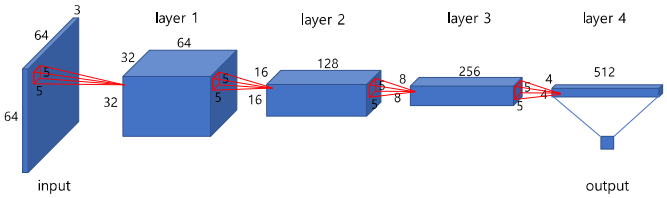

Fig. 7 Discriminator part of DCGAN model

IV. EXPERIMENT RESULT

For our experiment, we performed the k-means clustering on local image patches from SIFT, SURF, ORB, and the proposed method and visualized the clustered results through Fig. 8 to Fig. 13. The number of cluster center k is 20.

Fig. 8 Clustering result of SIFT

Fig. 9 Clustering result of SURF

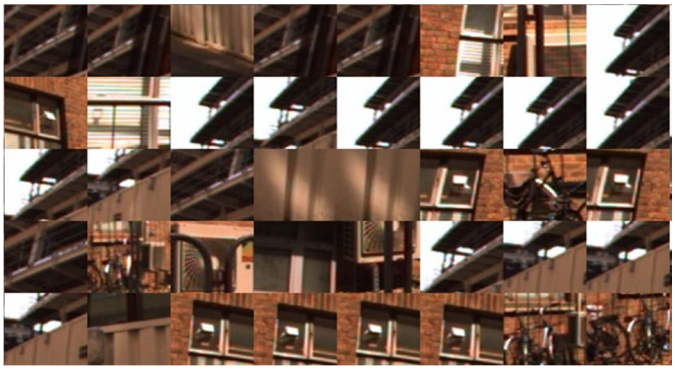

Fig. 10 Clustering result of ORB

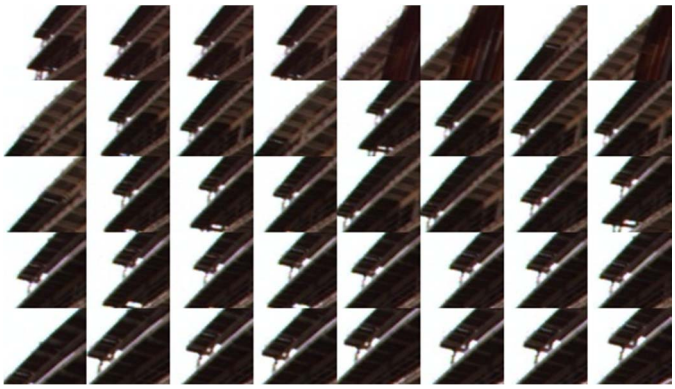

Fig. 11 Clustering result of the proposed method: class 1

As we can find from the clustering results, the local descriptors such as SIFT, SURF, and ORB badly gathered the patches with different appearances. However, the proposed method well aggregated local patches with similar appearances. In the future research, we will perform the bag-of-visual-word algorithm for the loop closure detection, combining with the proposed descriptor.

V. Conclusions

In this paper, we proposed a local patch descriptor using the deep convolutional generative adversarial network for loop closure detection in the simultaneous localization and mapping. The experiment results confirmed that the proposed method well cluster the local patches having similar appearances and it is better than the hand-crafted features like SIFT, SURF, and ORB. The important limitation lies in the fact that the proposed method requires an additional training phase for the local patch descriptor; however, it is practically acceptable since users exploit the estimation phase. The findings of this study suggest that the learning-based image descriptor will be essential primitives for the place recognition which is a ubiquitous challenge in the computer vision field. In that context, we will apply the proposed descriptor to a bags-of-visual-word algorithm for determining the loop closures in the simultaneous localization and mapping.

Fig. 12 Clustering result of the proposed method: class 2

Fig. 13 Clustering result of the proposed method: class 3

Acknowledgement

This work was supported by the 'Civil-Military Technology Cooperation Program' grant funded by the Korea government.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, “Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] D. Gálvez-López and J. D. Tardós, “Bags of binary words for fast place recognition in image sequences,” *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[3] Y. Hou, H. Zhang, and S. Zhou, “Convolutional Neural Network-Based Image Representation for Visual Loop Closure Detection,” in *IEEE International Conference on Information and Automation*, 2015, pp. 213–221.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *ImageNet Classification with Deep Convolutional Neural Networks*, 2012, pp. 1097–1105.

[5] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv*, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434.