

## Research Article

# A BigBiGAN-Based Loop Closure Detection Algorithm for Indoor Visual SLAM

Qiubo Zhong  and Xiaoyi Fang

Robots College, Ningbo University of Technology, Ningbo, Zhejiang 315211, China

Correspondence should be addressed to Qiubo Zhong; zhongqiubo@nbut.edu.cn

Received 4 March 2021; Revised 19 June 2021; Accepted 12 July 2021; Published 21 July 2021

Academic Editor: Vinod Sharma

Copyright © 2021 Qiubo Zhong and Xiaoyi Fang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Loop closure detection serves as the fulcrum of improving the accuracy and precision in simultaneous localization and mapping (SLAM). The majority of loop detection methods extract artificial features, which fall short of learning comprehensive data information, but unsupervised learning as a typical deep learning method excels in self-access learning and clustering to analyze the similarity without handling the data. Moreover, the unsupervised learning method does solve restrictions on image quality and singleness semantics in many traditional SLAM methods. Therefore, a loop closure detection strategy based on an unsupervised learning method is proposed in this paper. The main component adopts BigBiGAN to extract features and establish an original bag of words. Then, the complete bag of words is used to detect loop closing. Finally, a considerable validation check of the ORB descriptor is added to verify the result and output outcome of loop closure detection. The proposed algorithm and other compared algorithms are, respectively, applied on Autolabor Pro1 to execute the indoor visual SLAM. The experiment shows that the proposed algorithm increases the recall rate by 20% compared with ORB-SLAM2 and LSD-SLAM. And it also improves at least 40.0% accuracy than others and reduces 14% time loss of ORB-SLAM2. Therefore, the presented SLAM based on BigBiGAN does benefit much the visual SLAM in the indoor environment.

## 1. Introduction

The classic visual SLAM [1] system mainly includes four processes: front end, back end, loop detection [2–4], and mapping. The front end aims to (1) extract feature points from image sets, (2) calculate motion trajectory, and (3) establish initial mapping. And the back end concentrates on (1) optimizing the data provided by the front end and (2) eliminating the noise in real-time trails. Simultaneously, loop closure detection could cut down accumulative errors or drifted outcomes in the progression. Accumulative errors could increase throughout localization and mapping owing to the thoughtless comparison of the current frame and its adjacent frame. It could result in (1) worse computation estimation and (2) the unserviceable constructed tracking map. Thereby, loop closure detection plays a significant part in the visual SLAM framework to effectively reduce accumulative errors and feature redundancy as much as possible

by detecting overlapping features. Inevitably, the number of comparison frequencies must swell in the process of loop closure detection, which could bring about (1) large computation requirement, (2) inefficient time-consuming situations, and (3) a glut of repetitive scenes of visual SLAM. Recently, the research on loop closure detection has made valuable progress in addressing the above difficulties.

*1.1. Loop Closure Detection.* A descriptor was designed in [5] for the overall image, which improves the correlation of image calculation and efficiency of loop detection. Jia Song-Min et al. [6] improved loop detection accuracy by combining the traditional visual SLAM with random image coding of fern color. Recently, with the rapid development of neural networks in computer vision, there are many kinds of researches having made great achievements on combining visual SLAM with neural networks. Gao and Zhang [7]

constructed a dictionary using the CNN (convolutional neural network) to extract image features, which was different from the traditional bag-of-words model, and achieved a more accurate effect in loop detection. Bai et al. [8] made full use of intermediate convolution layers to reduce the range of searching, which improved the efficiency of loop detection. Zhang et al. [9] preprocessed the extracted dataset before using the neural network to calculate the final results, and the entire detection process had high efficiency. Zhang [10] proposed an unsupervised deep structure of the generative adversarial network (GAN) to detect features instead of using artificial ones, which performed well in the outdoor environment. Li et al. [11] proposed a two-image fusion method which used RGB-D images in feature extraction and convolutional neural networks in loop detection to get higher accuracy and real-time capability. Xu and Bu [12] also presented an original approach on loop detection for improving accuracy and efficiency. Firstly, it detected image sequences with a preprocessed Faster-RCNN and constructed landmark 2D semantic feature vector graphs with the former detected outputs, including feature types, image semantic, and pixel positions. Then, the non-linear accumulative error was used to obtain the initial value of loop closure and calculate the similarity between different 2D semantic vector graphs. Lastly, the final loopback results came out after position verification.

The unsupervised semantics of neural networks might not perform in visual SLAM and function the end-to-end advantages as expected owing to indeterminate features in networks and computation errors. Although some could obtain high precision, they did increase the frequency of comparison over time.

**1.2. Generative Adversarial Networks.** The unsupervised learning method can extract feature structures and construct a task model from the initial data without additional labels. In recent years, generative adversarial network [13] (GAN) proposed by Ian Goodfellow in 2014 has made notable achievements in the field of unsupervised learning. In 2017, DeepMind added an encoder that maps data to the hidden feature space based on the GAN called BiGAN [14], which gives GAN the ability to learn represented features. In 2018, DeepMind made an improvement on GAN [15], which could effectively train large-scale GAN and, meanwhile, improve BiGAN's generator. In 2019, the presented BigBiGAN [16] that has almost the same structure and theory as BiGAN, combining BiGAN essential abilities of representation learning and the large-scale training network, effectively advanced its discriminator. Thus, the strength of BigBiGAN is adopted to apply in SLAM to distinguish loop closing.

In this paper, the proposed algorithm uses BigBiGAN to extract features and construct the compatible bag-of-words model instead of the traditional handcrafted features, with combining extracted features with valuable characteristics of the classic bag-of-words model. Sequentially, the unsupervised learning-based dictionary is applied to detect loop closures in visual SLAM. Meanwhile, a considerate

verification is added to further assure the validity of loop closure detection by using the ORB descriptor before obtaining the ultimate. The proposed algorithm and other compared algorithms are, respectively, applied on Autolabor Pro1 to execute the indoor visual SLAM. And the validity, accuracy, and recall rate are taken into comparison analysis. The experiment shows that the presented algorithm performs well in avoiding the shortcomings of using artificial image features and improving the (1) detection accuracy, (2) detection efficiency, and (3) time-consuming issue through the generative adversarial model's learning results.

## 2. Materials and Methods

**2.1. Implementation of the Bag-of-Words Model.** Figure 1 shows the structure of BigBiGAN.  $x$  denotes the real data (image), and  $z$  denotes the reconstruction of the data  $x$ . Discriminator  $D$  includes three parts. The right part distinguishes the generated data from the original data. The left part judges whether  $z'$  extracted by the encoder is the same as  $z$  input to the generator. Both units are aimed to ensure that the optimization process could move towards the optimal solution. The bottom part combines  $x$  and  $z$ . In order to calculate loss, the encoder [16] part (1) and decoder [16] part (2), respectively, compose  $D$  and  $G$ . The loss part utilizes (3) and (4) to acquire the optimizing result.

Joint (probability) distribution of the encoder is

$$q(x, z) = q(x)q(z | x). \quad (1)$$

Joint distribution of the decoder is

$$p(x, z) = p(z)p(x | z). \quad (2)$$

The objective function of BiGAN is as follows:

$$\min_{G, E} \max_D V(D, E, G). \quad (3)$$

Converting (3) into a mathematical formula,  $V$  is defined as

$$\begin{aligned} V(D, E, G) &:= E_{x \sim p_x} [E_{z \sim p_E(\cdot|x)} [\log D(x, z)]] \\ &\quad + E_{z \sim p_z} [E_{x \sim p_G(\cdot|z)} [\log(1 - D(x, z))]] \\ &= \log D(x, E(x)) + \log(1 - D(G(z), z)). \end{aligned} \quad (4)$$

The ORB bag-of-words model can be expressed as

$$\begin{aligned} D(O) &= \sum orb, \\ D(O) &= \{O_1, O_2, \dots, O_n\}. \end{aligned} \quad (5)$$

The bag-of-words model constructed in this paper can be expressed as a mathematical formula as follows:

$$\begin{aligned} D(B) &= \sum \text{BigBiGAN}, \\ D(B) &= \{B_1, B_2, \dots, B_n\}. \end{aligned} \quad (6)$$

The proposed Algorithm 1 adopted BigBiGAN to generate the bag-of-words model. (1) The prepared real scene

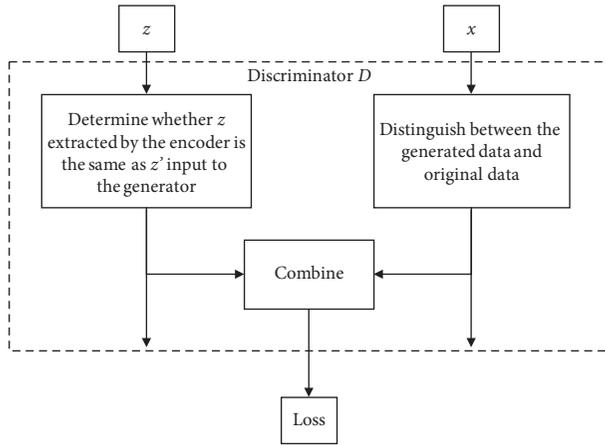


FIGURE 1: The structure of BigBiGAN.

pictures are input into the neural network in advance, and the model outputs two sets of features that are sequentially provided by the discriminator and generator. (2) Take these two types of feature sets as a core judgment dictionary and an auxiliary dictionary. The implementation of the bag-of-words model mainly includes the construction of feature dictionaries and calculation of similarity, whose pseudo-code steps are given as follows.

Furthermore, the training results of the neural network concern more with high-level semantics rather than pixel-level details, which could help to filter the similarity of pixels with the trained model and then efficiently improve the algorithm's robustness. The utilization of the pretrained BigBiGAN [16] model for feature extraction takes advantage of adversarial network comprehension of the entire semantic pictures rather than traditional diversity detection in the pixel level. The strategy makes it possible that the loopback detection equipped with an unsupervised representation learning network could have reliable performance in a diverse environment.

**2.2. Implementation of Main Components.** Based on the above analysis, the steps of the designed algorithm framework in this paper are as follows:

Step 1: take images of scenes and select keyframes as the input of the neural network while the vehicle is driving.

Step 2: extract the feature map from the input keyframes through the neural network.

Step 3: compare the extracted features with the features in the real-time discriminant network dictionary. If a loop closure is detected, go to Step 4. Otherwise, finish.

Step 4: compare the extracted features with the features in the real-time generated network dictionary. If a loop closure is detected, go to Step 5. Otherwise, finish.

Step 5: verify the loop closure image. If the loop closure is correct, go to Step 6. Otherwise, finish.

Step 6: output the matched result of loop closure detection and adjust the camera pose.

The purpose of loop detection is to optimize the pose of the camera and eliminate accumulated errors. The algorithm uses a generative adversarial network to improve the accuracy of detection and eliminate the large errors in the detection results. Meanwhile, it faced a challenge that two similar images in the loop closure state could not be differed by the neural network owing to similar feature maps of semantic features and feature pixel locations.

Considering the above challenge of eliminating false similarity detection, the algorithm adopts the ORB descriptor to verify the results of the loop closure detection. As shown in Figure 2, there exist high similarity and symmetry in two photos taken by using a camera with different poses. The small circles and connection lines, respectively, denote ORB feature points and the matching of the feature points. ORB feature points have good rotation invariance. In this case, the adoption of camera pose verification can effectively eliminate false detections.

The pseudo-code of the algorithm in this section is given in Algorithm 2.

**2.3. Experiment.** In this paper, the experimental hardware environment mainly uses Autolabor Pro1. Autolabor Pro1 is a navigation car focused on unmanned driving. Figure 3 shows the car. The car has RGB-D camera Kinect V2, electric screen, computation manager, several buttons, four normal wheels, and the yellow bottom, which could be easily controlled and developed by the ROS.

The software environment includes Ubuntu 16.04, TensorFlow, and Python/C++. Some real images and part of features are shown in Figure 4. An unmanned mobile car is equipped with a prephotographed camera, which could provide 150 binary images in  $960 \times 540$  (dpi).

The comparative experiment algorithm is ORB-SLAM2 [17, 18], which is a real-time visual SLAM system that uses ORB feature points in Figure 5 as visual odometers and feature points in loop closure detection. The ORB-SLAM2 system also supports monocular, binocular, and depth cameras and has good performance indoor or outdoor. The system has good rotation invariance and is one of the better-performing methods.

Additionally, the experiment selects LSD-SLAM [19] as the second comparison algorithm. LSD-SLAM is a direct method in visual SLAM, whose loop detection method still uses the feature point calculation mode. Therefore, this algorithm could provide a meaningful reference in the comparison of loop detection.

### 3. Results and Discussion

**3.1. Validity Analysis.** As shown in Figure 6, the real trajectory of the field in this experiment presents a closed loop with the start and end overlapping. The trajectory is

```

Input: RGB images
Output: BoW
(1) FOR  $i = 1$  to  $imgCollection.Length()$  DO
(2)    $img \leftarrow imgCollection[i]$ 
(3)    $B \leftarrow BigBiGAN(img)$ 
(4)    $BoWG.add(B.feature(Generative))$ 
(5)    $BoWD.add(B.feature(Discriminator))$ 
(6) END FOR

```

ALGORITHM 1: BoW component.

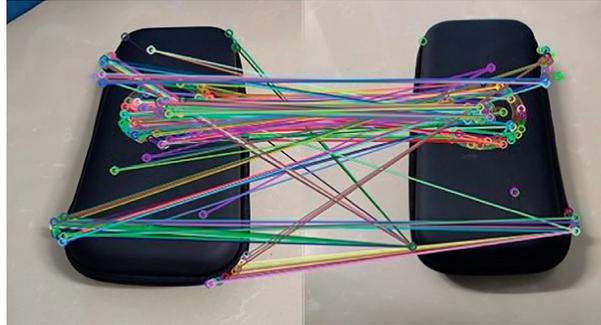


FIGURE 2: The case of similar photos.

```

Input: RGB image
Output: a loop closure detection result
(1)  $img \leftarrow GetRGBgray( RGB)$ 
(2)  $x \leftarrow FeatureExtraction(img)$ 
(3) FOR  $i = 1$  to  $BoWD.Length()$  DO
(4)   IF  $x == BoWD[i]$  THEN
(5)      $D \leftarrow TRUE$ 
(6)   ENDFOR
(7)   IF  $D$  THEN
(8)     FOR  $i = 1$  to  $BoWG.Length()$  DO
(9)       IF  $x == BoWG[i]$  THEN
(10)         $G \leftarrow TRUE$ 
(11)      END FOR
(12)    ELSE
(13)      RETURN FALSE
(14)    END IF
(15)    IF  $G$  THEN
(16)       $L \leftarrow ORBCheck(imgCollection[i], img)$ 
(17)    ELSE
(18)      RETURN FALSE
(19)    END IF
(20)    IF  $L$  THEN
(21)      RETURN TRUE
(22)    ELSE
(23)      RETURN FALSE
(24)    END IF

```

ALGORITHM 2: Loop closure main component.

composed of a straight line and a curve. Figure 6(a) shows the range of the real trajectory's width and length. Figure 6(b) shows the trajectory in the top view. Figure 6(c) shows the trajectory in the oblique view.

The experiment used the ORB-SLAM loop closure detection, the LSD-SLAM loop closure detection, and the proposed loop closure detection method to pose and point cloud recovery. There are four results in Figures 7(b)–7(d),

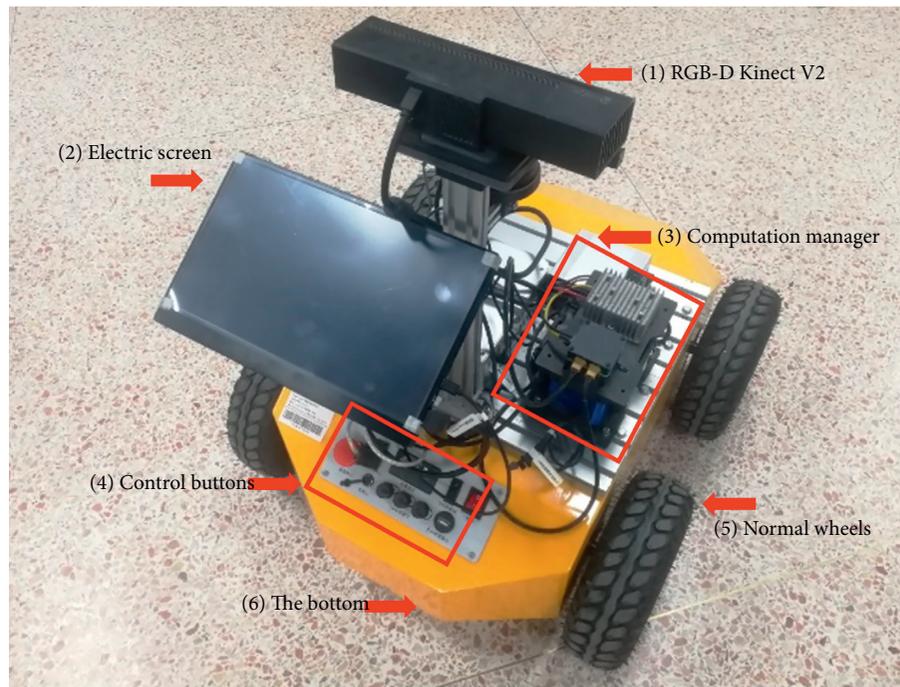


FIGURE 3: The unmanned car of Autolabor Pro1.

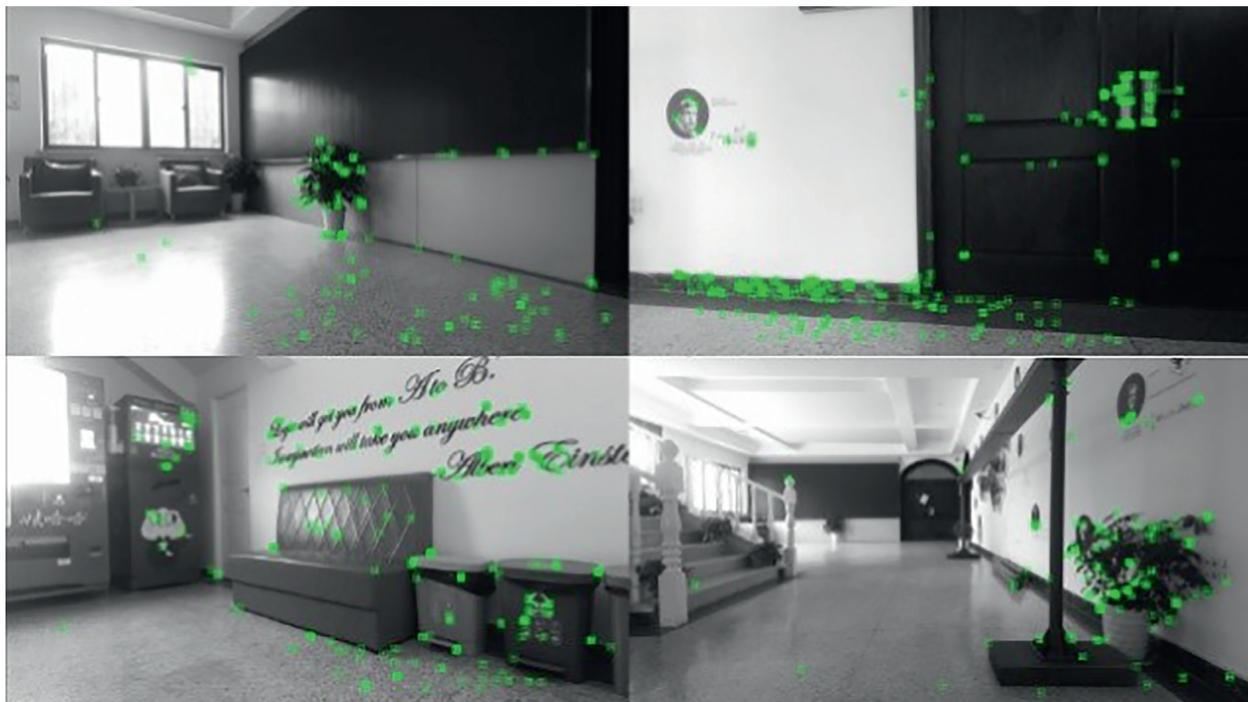


FIGURE 4: Part of the dataset.

where Figure 7(a) did not use loop closure detection algorithm and Figures 7(b)–7(d), respectively, used the above three algorithms.

Different methods of eliminating accumulated errors in three algorithms perform three pretty different constructed maps in the same experimental scene. The accumulated error

kept increasing in the algorithm that did not use the loop closure detection algorithm. It is obvious to find the drift phenomenon in Figure 7(a) where the generated trail is significantly different from the initial trajectory. According to the results of ORB-SLAM2 in Figure 7(b) and LSD-SLAM in Figure 7(c), there are still many errors in the recovered

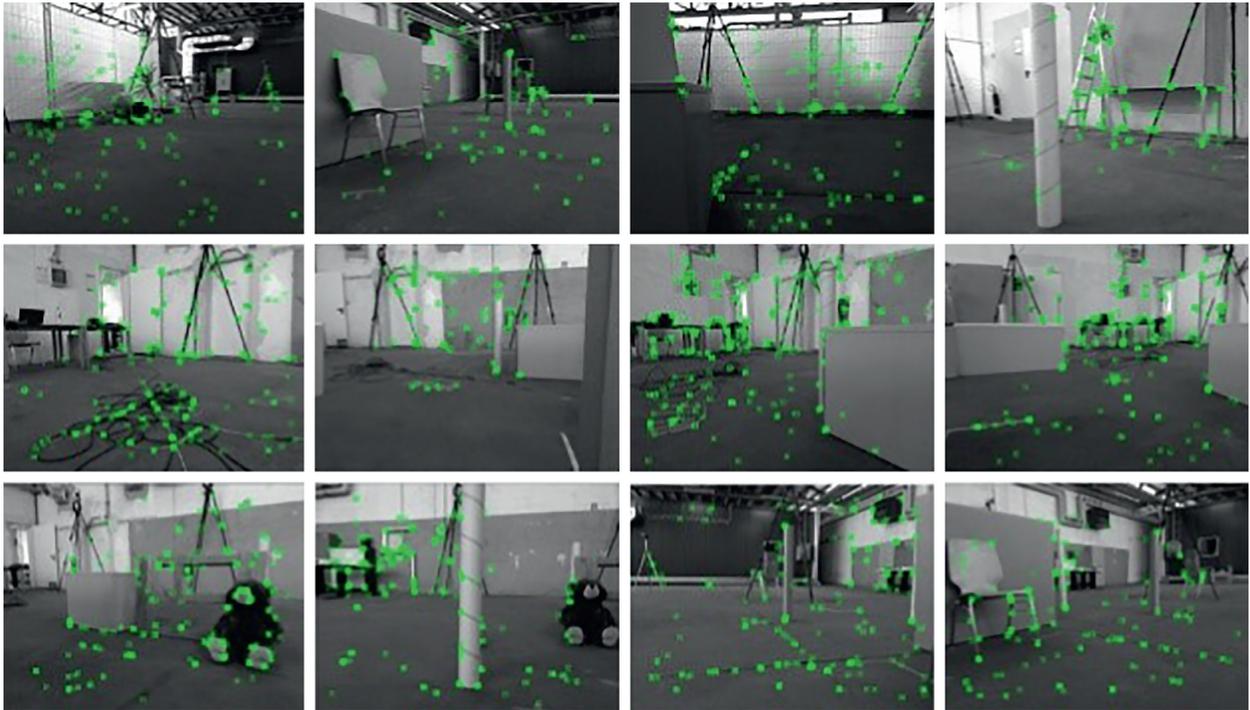
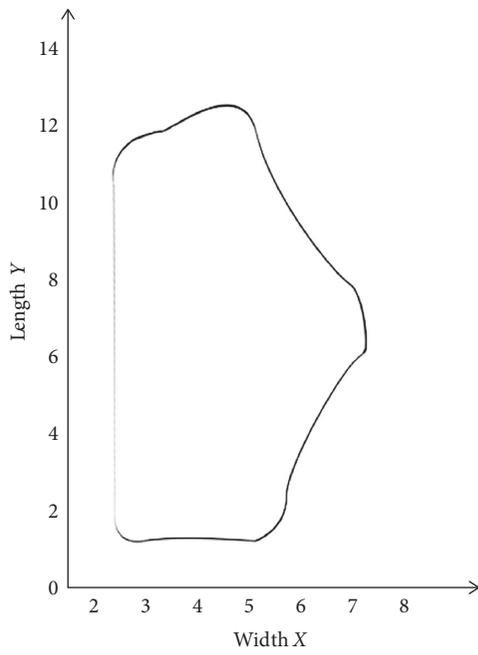
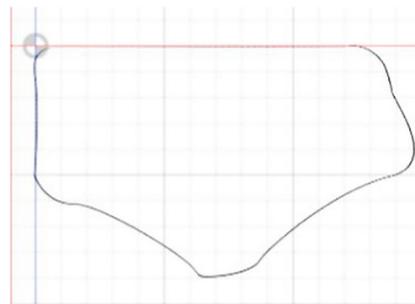


FIGURE 5: ORB feature extraction result.



(a)



(b)

FIGURE 6: Continued.

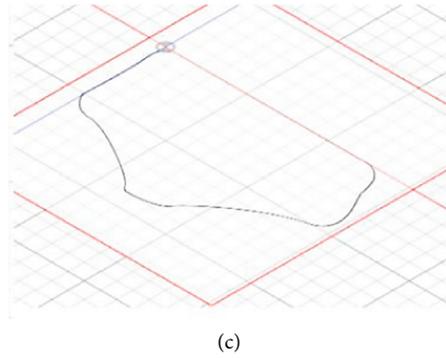


FIGURE 6: Top view and oblique top view of the trajectory. (a) The real trajectory. (b) Top view. (c) Oblique view.

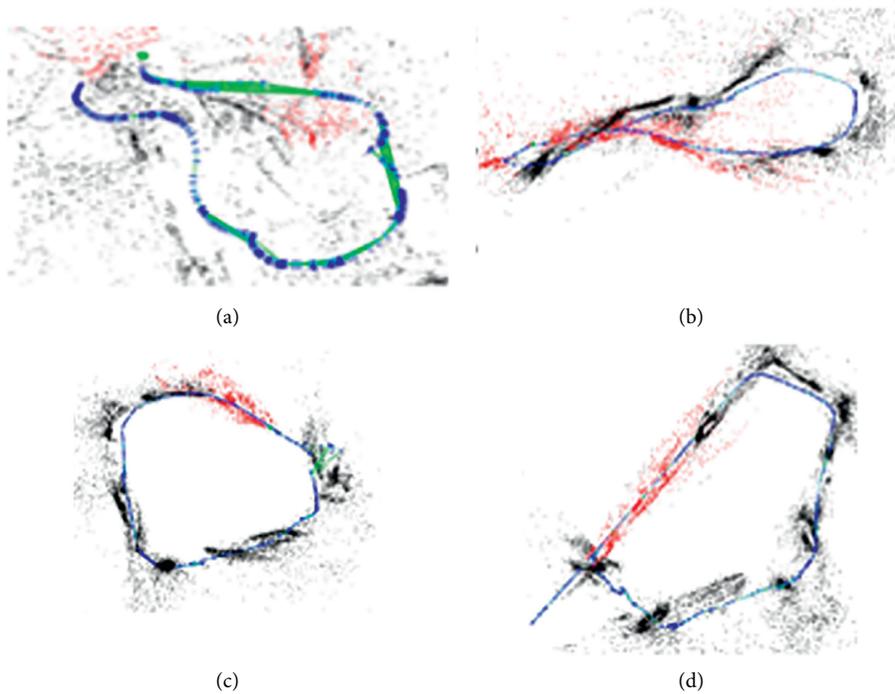


FIGURE 7: Pose recovery and point cloud recovery results of four loop detection algorithms: (a) recovery 1, (b) recovery 2, (c) recovery 3, and (d) recovery 4.

trajectory. The former had a length of the overlapping trail and an incomplete recovery. Although the latter showed a closure trail with few repeated recovery points, its output recovery did not match the real trajectory. Especially, it performed not as expected in the section of the curve trail. Notably, the recovery in Figure 7(d) was the most similar trail to the real trajectory in terms of the straight line and the curve. Thus, by utilizing the proposed loop closure detection, the visual SLAM system could correctly identify that the vehicle has passed the same position and adjusted the drift caused by the initial posture trajectory [20].

The experimental data obtained from the proposed algorithm are shown in Figure 8. Each curve denotes one result of the object compared to the number of frames. The  $x$ -axis represents the frame number ranging from 120 to 150.

The  $y$ -axis represents the similarity ranging from 0 to 100. For example,  $i7$  to  $i10$  on  $x$ -axis frame 148, respectively, have performed similarity analysis, but the data show that the similarity to  $i10$  is the highest, which is the most similar and judged as a loopback.

Hence, the point cloud recovery and posture trajectory perform more precisely, which proved the validity of the proposed method.

**3.2. Accuracy Analysis.** The accuracy of loop closure detection includes accuracy rate and recall rate. A high recall rate could result in a low accuracy rate, and vice versa. There are five pairs of loops:  $i1$  with frame 137,  $i5$  with frame 140,  $i7$  with frame 144,  $i9$  with frame 147, and  $i10$  with frame 148. In

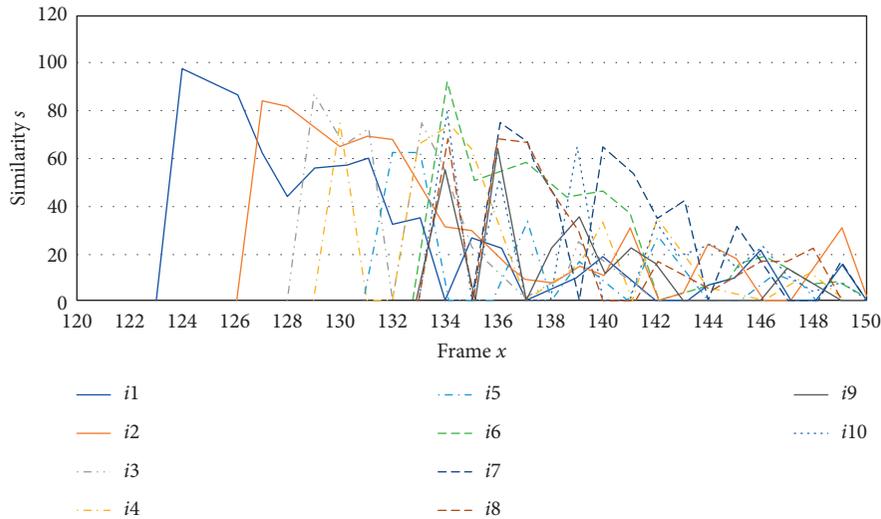


FIGURE 8: Similarity comparison result analysis.

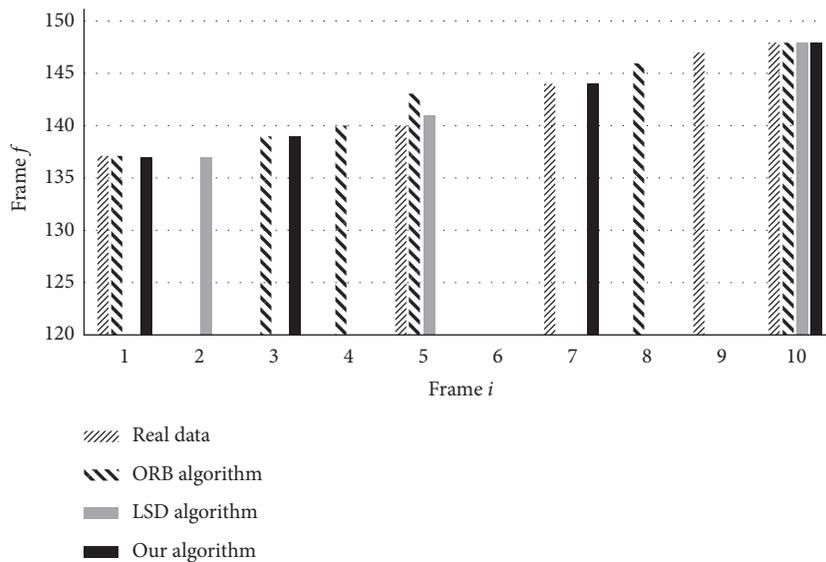


FIGURE 9: Comparison of loopback test results.

Figure 9, the horizontal coordinate indicates the first ten sheets in the dataset, and the vertical coordinate indicates the image order number of the corresponding dataset detected as a loop by the algorithm, which is the value of the histogram.

The ORB algorithm detected six loops that, respectively, existed in  $i1$  with frame 137,  $i3$  with frame 139,  $i4$  with frame 140,  $i5$  with frame 143,  $i8$  with frame 146, and  $i10$  with frame 148, but these six loops only included two correct loops. Two-sixth loops were accurate. The LSD algorithm detected three loops that, respectively, existed in  $i2$  with frame 137,  $i5$  with frame 141, and  $i10$  with frame 148. One-third loops were correct. The accuracy rate in the experiment was 33.3%. As for the proposed algorithm, it detected three-fourth real loops.

A more detailed comparison statistic of detection results among the LSD algorithm, ORB algorithm, and the proposed algorithm is shown in Table 1. The recall rate is the probability of being correctly detected in all real loops.

The proposed algorithm performed a 75% accuracy rate and a 60% recall rate, compared to the ORB with a 40% recall rate and the LSD with a 40% recall rate. From the statistical result, although it did not detect all existed loops, the proposed algorithm has a relatively higher recall rate and a precise rate than the LSD detection method and the ORB detection method.

In the paper, integrating an advantage of classic semantic features such as positions and angles, the proposed algorithm targets much more feature comprehension at the analysis of the similarity between images. Furthermore, the

TABLE 1: Analysis of the detection results of three algorithms.

Algorithm	Number of detection loops	Number of correct loops	Number of real loops	Accuracy (%)	Recall rate (%)
ORB	6	2	5	33.3	40
LSD	3	1	5	33.3	20
This article	4	3	5	75	60

TABLE 2: Time-consuming analysis of three algorithms.

Algorithm	Feature point detection/neural network analysis (s)	Similarity calculation (s)	Verification (s)	Total time (s)
ORB	61.8	13.2	1.8	76.8
LSD	58.7	14.6	1.5	74.8
This article	52.3	12.1	1.6	66

algorithm takes full use of the available data and becomes more compatible to accommodate the unfamiliar scene. Therefore, the presented algorithm has a higher accuracy and recall rate than other bag-of-words algorithms.

**3.3. Efficiency Analysis.** Three time-consuming sections, feature detection, similarity calculation, and verification, showed detailed time cost about the LSD, ORB, and proposed algorithm in Table 2. The similarity calculation process in this paper and the ORB method uses the bag-of-words model. The essence of these three methods features in point comparison, so these three methods consumed similar time. The proposed algorithm cost 52.3s in feature point detection part and 12.1s in similarity calculation part, both of which behave superior to other two methods. And the LSD method costs the least time 1.5s among all methods on verification part. The proposed algorithm reduced the total time by 14% compared to the ORB and dropped it by 11.7% compared to the LSD.

In conclusion, compared to loop closure detection in ORB-SLAM2 and LSD-SLAM, the proposed algorithm accuracy increased by at least 40% based on a 20% increase in recall rate, and the total time consumption reduced by at least 11%.

## 4. Conclusion

This paper proposes a composed loop closure detection algorithm based on an adversarial model BigBiGAN, which improves the accuracy and efficiency of loop detection. Then, the proposed algorithm was applied in the real indoor visual SLAM. The experimental results showed that the algorithm achieved higher accurate loop closure detection than the other two methods. In terms of time consumption, the visual SLAM framework integrating the proposed method also outperformed the other two. Nevertheless, the adaptability of the algorithm is deeply affected by the unsupervised model of learning representation feature

extraction. Hence, large-scale datasets and more experiments are expected to strengthen and advance the presented loop closure detection algorithm in indoor visual SLAM.

## Data Availability

The authors collected data of the scenes in our daily environment to train the BigBiGAN model, and the final verification was also in the scene. There is still room for improvement in this algorithm and a data, which is not ready to be provided at present.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] C. Cadena, L. Carlone, H. Carrillo et al., "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [2] K. A. Tsintotas et al., "Appearance-based loop closure detection with scale-restrictive visual features," in *Proceedings of the International Conference on Computer Vision Systems*, Springer, Thessaloniki, Greece, September 2019.
- [3] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "Modest-vocabulary loop-closure detection with incremental bag of tracked words," *Robotics and Autonomous Systems*, vol. 141, Article ID 103782, 2021.
- [4] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proceedings of the Computer Vision, IEEE International Conference on. Vol. 3. IEEE Computer Society*, Madison, Wisconsin, June 2003.
- [5] L. Bampis, A. Amanatiadis, and A. Gasteratos, "Fast loop-closure detection using visual-word-vectors from image sequences," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 62–82, 2017.
- [6] 贾. Jia Song-Min, 李. Li Bo-Yang, and 张. Zhang Guo-Liang, "TSDF map building based on hybrid loopback detection and

- closed-loop optimization,” *Optics and Precision Engineering*, vol. 26, no. 6, pp. 1497–1506, 2018.
- [7] X. Gao and T. Zhang, “Loop closure detection for visual SLAM systems using deep neural networks,” in *Proceedings of the 34th Chinese Control Conference*, pp. 5851–5856, IEEE Press, Piscataway, NJ, USA, July 2015.
- [8] D. Bai, C. Wang, Bo. Zhang et al., “CNN feature boosted SeqSLAM for real-time loop closure detection,” *Chinese Journal of Electronics*, vol. 3, 2018.
- [9] X. Zhang, Y. Su, and X. Zhu, “Loop closure detection for visual SLAM systems using convolutional neural network,” in *Proceedings of the 23rd IEEE International Conference on Automation and Computing*, pp. 1–6, IEEE Press, Piscataway, NJ, USA, September 2017.
- [10] K. Zhang and W. Zhang, “Loop closure detection based on generative adversarial networks for simultaneous localization and mapping systems,” *Chinese Automation Congress (CAC)*, vol. 2017, pp. 7916–7919, Article ID 8244215, Jinan, 2017.
- [11] R. Li, G. Xu, R. Guo et al., “Research on RGB-D SLAM loop detection based on convolutional neural network,” *Journal of Heilongjiang Institute of Technology (Integrated Edition)*, vol. 19, no. 11, pp. 42–48, 2019.
- [12] J. Xu and F. Bu, “Optimization algorithm of loop detection based on Faster-RCNN,” *Computer Application Research*, vol. 36, no. 12, pp. 3628–3631, 2019.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” *Communications of the ACM*, In NeurIPS, vol. 63, no. 11, 2014.
- [14] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” ICLR, 2017, <https://arxiv.org/abs/1605.09782>.
- [15] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” ICLR, 2019, <https://arxiv.org/abs/1809.11096>.
- [16] J. Donahue and K. Simonyan, “Large scale adversarial representation learning,” *Advances in Neural Information Processing Systems*, vol. 19, pp. 10541–10551, 2019.
- [17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [18] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras,” 2016, <https://arxiv.org/pdf/1610.06475.pdf>.
- [19] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: large-scale direct monocular SLAM,” *Computer Vision-CCV 2014 Lecture Notes in Computer Science*, vol 8690, pp. 834–849, Springer, Cham, Switzerland, 2014.
- [20] H. Alismail, B. Browning, and S. Lucey, “Photometric bundle adjustment for vision-based SLAM,” in *Proceedings of the Asian Conference on Computer Vision*, pp. 324–341, Springer, Perth, Australia, December 2016.