sustainability

Article

Deep Pose Graph-Matching-Based Loop Closure Detection for Semantic Visual SLAM

Ran Duan , Yurong Feng and Chih-Yung Wen *

Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China
* Correspondence: chihyung.wen@polyu.edu.hk

Abstract: This work addresses the loop closure detection issue by matching the local pose graphs for semantic visual SLAM. We propose a deep feature matching-based keyframe retrieval approach. The proposed method treats the local navigational maps as images. Thus, the keyframes may be considered keypoints of the map image. The descriptors of the keyframes are extracted using a convolutional neural network. As a result, we convert the loop closure detection problem to a feature matching problem so that we can solve the keyframe retrieval and pose graph matching concurrently. This process in our work is carried out by modified deep feature matching (DFM). The experimental results on the KITTI and Oxford RobotCar benchmarks show the feasibility and capabilities of accurate loop closure detection and the potential to extend to multiagent applications.

Keywords: pose graph; loop closure detection; semantic VSLAM; deep feature matching

Citation: Duan, R.; Feng, Y.; Wen, C.-Y. Deep Pose Graph-Matching-Based Loop Closure Detection for Semantic Visual SLAM. *Sustainability* **2022**, *14*, 11864. https://doi.org/10.3390/su141911864

Academic Editors: Yushuai Li, Ning Zhang and Jiayue Sun

Received: 9 August 2022
Accepted: 12 September 2022
Published: 21 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Loop closure detection or subsequent relocalization is a critical process for the visual simultaneous localization and mapping (SLAM) system [1]. This is an event-triggered process [2–4] that can eliminate the cumulative error of navigation trajectory estimation by detecting the locations visited by the robot and correcting the estimated odometry. For intelligent autonomous mobile robots, vision-based mapping, navigation, and localization are fundamental tasks, and the VSLAM system integrates all of them. VLSAM is an active research topic in the field of computer vision and robotics, especially for GNSS-denied environments, such as urban areas, indoor rooms, or underground spaces [5]. The most common solution to the visual navigation system is the VSLAM [6–10]. We illustrate a general VSLAM system in Figure 1. A modern VSLAM system consists of two major modules. Incremental estimation of the motion by visual odometry is usually the task of what is called the frontend. Quick elimination of the cumulative error by optimization is usually the responsibility of the backend of the system [11,12]. There are two problems to be solved in the frontend: reconstructing 3D scenes and estimating the camera pose in the current frame [13]; however, they are dependent upon each other. The 3D reconstruction requires the known poses of the camera, while the estimation of the absolute pose of the camera (with real distance) requires the known 3D scene. The basic framework to solve those two problems concurrently is the structure from motion, as well as visual odometry, in which we focus on the pose estimation for navigation [14,15]. With a depth sensor or stereo view, the coarse 3D scene can be generated at the beginning. Then, the fundamental steps for the frontend are tracking the image scenes (feature points, landmarks, objects [16–21], etc.) between frames and estimating the camera motion with the geometry constraints. The backend records key information from the frontend, such as the 3D scene, landmarks, and geometry constraints in keyframes over the navigation, and registers them into a global map. With that information, the backend refines the estimation results by feeding a set of keyframes, i.e., the pose graph, into the optimizer [22]. Meanwhile, when the frontend detects similar scenes, and the backend believes the UAV has returned to a known place in the global map, the loop closure process will be activated. Loop closure detection may be either online or offline, while optimization may only be conducted offline.

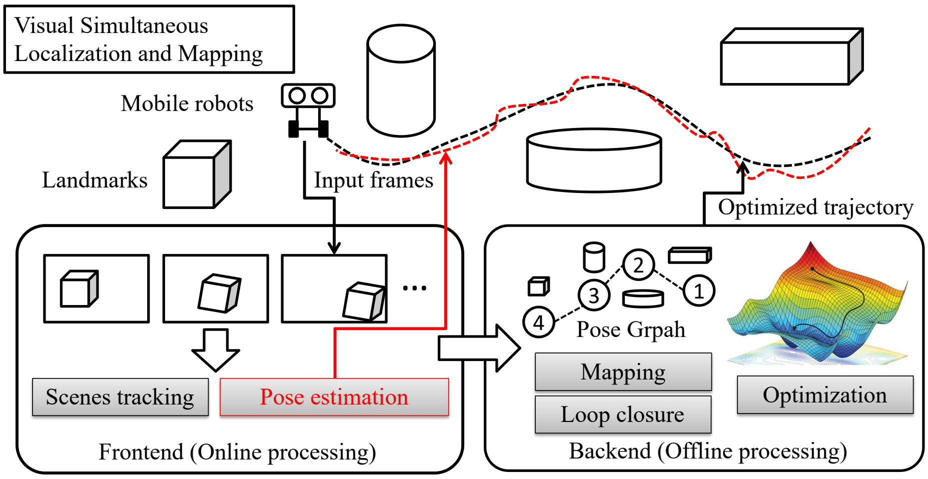

Figure 1. Visual simultaneous localization and mapping.

The most challenging problem for VSLAM is dealing with the cumulative error. The cumulative error is usually calculated by measuring the root mean square pose error between the estimates and the provided ground truth, which is a type of relative error. As the odometry distance increases, the tiny errors from the frame-to-frame estimations become significant. In particular, rotation errors may be amplified over long-term navigation, which can induce large localization errors. Cumulative errors mainly come from the frontend, i.e., the structure-from-motion process (SfM), in which the robot estimates its camera motion and recovers the 3D scene structure frame by frame [23]. The mathematical model of this process has been well studied in computer vision, projective geometry, and photogrammetry [24,25]. The obstacles to implementing this theory in robotic applications are the constraints in the real world. Keypoint detection and tracking inevitably introduce noise and outliers. To deal with these problems, researchers have designed accurate and robust feature descriptors, such as ORB-SLAM [26], or have estimated the motion by directly minimizing the residual of images, such as DTAM [27] and LSD-SLAM [28]. Whether feature-based or direct methods, the idea is to form an algebraic system, which is usually nonlinear and overdetermined with noise and outliers, and find the optimal solution. However, the local minimum problem during the regression or optimization for solving such a system is always a critical issue for obtaining accurate estimation results. Although many optimization methods have been proposed to remove the outliers and reduce the impact of data noise, and the frame-to-frame estimation error has been minimized to a trivial residual [29,30], the VSLAM system still suffers from a cumulative error and the occasional wrong estimation. At the top system level, IMU-aided sensor fusion in conjunction with a Kalman filter (KF) [31], bundle adjustment (BA) [32], or pose graph optimization [33,34] may be the solution. The optimization problems formed from BA or pose graphs are usually complex and non-convex. Although many optimizers have been proposed [35–41] for solving these types of problems, it is rather difficult for the system itself to be aware of the drifts [42]; open loop estimation and loop closure detection are very likely to be the only reliable way to eliminate the cumulative error for VSLAM. However, this process requires accurate and efficient image retrieval and matching.

This paper addresses the loop closure detection problem using deep learning-based pose graph matching. There are two key steps in loop closure detection: (1) matching the current scenes with the recorded keyframes and (2) finding the transformation between the matched keyframes for trajectory correction. Because a long-term VSLAM generates a large number of keyframes, the image views of the keyframes are usually discretized using feature descriptors. After the keyframes are matched, the correction of the trajectory is carried out by pose graph optimization. In our work, we combine the keyframe descriptors and pose graphs by modeling them into a sparse image, for which each keyframe is converted into a feature point. Then, we employ a deep convolutional neural network (CNN)-based method, i.e., deep feature matching (DFM) [43], to solve the two key steps simultaneously over a large data scale. The proposed method is evaluated on two self-driving car benchmarks: the KITTI [44] and Oxford RobotCar datasets [45]. An illustration of the main idea is shown in Figure 2. The main contributions of this work can be summarized as follows:

- To the best of our knowledge, this is the first time that a sparse image-based keyframe map that stores each keyframe as a feature point with convolutional feature descriptors is proposed for pose graph matching.
- We convert the loop closure detection problem to a feature point matching problem so that keyframe matching can be performed over a large data scale with a geometry transform consensus.
- We evaluate the method on the KITTI and Oxford RobotCar benchmarks, which demonstrates the feasibility of the proposed method and the potential for its application in multiagent robotics.

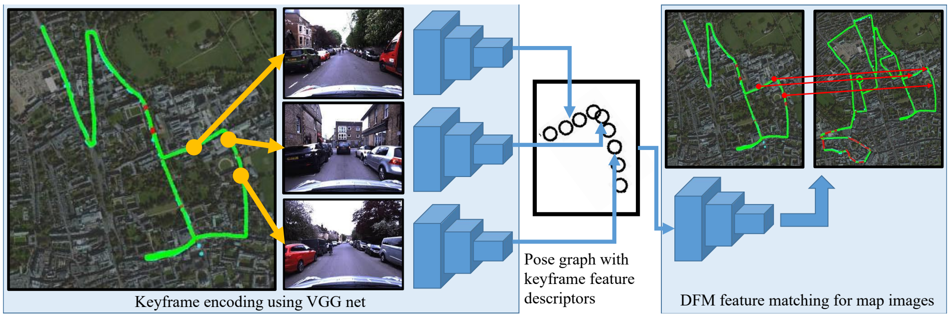
Keyframe encoding using VGG net
Pose graph with keyframe feature descriptors
DFM feature matching for map images

Figure 2. Conversion of the loop closure detection problem to image feature matching. The green lines represent estimated trajectories. The yellow dots represent the recorded keyframes in database. Red lines are the transformation between matched keyframe and current keyframe.

2. Literature Review

Traditional loop closure detection methods usually encode the keyframe data using a set of feature point descriptors, such as the bag-of-words (BoWs) [46,47] or a principal component analysis (PCA). This means the image retrieval process for loop closure detection has been basically treated as an image classification problem. For example, the vector of locally aggregated descriptors (VLAD) aggregates the local descriptors, i.e., the BoWs, into a compact global image representation. The Fisher vector, on the other hand, uses the Gaussian mixture model (GMM) to create a visual word dictionary matching and hence encodes more visual information than the BoW. Except for 2D image-based methods, some SLAM systems detect the overlap locations by matching the reconstructed 3D map [48,49]. However, 3D point cloud alignment with unknown overlap areas is a highly complex problem mathematically.

Recently, with the development of deep learning [50], semantic VSLAM has attracted a lot of interest. With regard to the pure vision SLAM system, recent efforts have been made in semantic scene tracking, which is one level up from the feature point tracking during the frontend stage. Meanwhile, deep learning-based feature point detection and matching methods are based on a similar idea. For instance, SuperGlue, SuperPoint, DFM, etc., use different layer outputs of CNN features as the point descriptors. These methods have been used for both camera motion estimation and loop closure detection. A notable work, SuperGlue [51], proposed the attentional graph neural network (AGNN). SuperGlue uses a five-layer multilayer perceptron (MLP) to encode the feature descriptors with locations, for which the relationship between neighbor feature points is trained by the attentional aggregation scheme. Then, it finds the matched feature points between two images using an optimal matching layer. In their subsequent work, SuperPoint [52], the authors proposed a

MagicPoint Base Detector net and homographic adaptation for deep feature self-supervised learning. In [43], a two-stage feature point matching DMF-net was presented. It uses a dense nearest neighbor search (DNNS) to find the best match for each element between two deep feature encoded images. To address the local illumination variation problem, the authors proposed the hierarchical refinement algorithm (HRA), which changes the number of feature points in each re-refinement stage by selecting the appropriate matching values and excluding outliers. All of these methods use CNN as the point feature encoder, and it is clear that loop closure detection and image point matching are fundamentally similar. In fact, the feature semantic level in the loop closure detection should be higher than the point matching because the task is to match the keyframes between two pose graphs [53]. However, most semantic SLAM systems still perform loop closure detection by image retrieval in feature point or image patch levels.

The semantic features used in our work move one level up. We treat the whole navigation map as the global image and each keyframe as the feature point. Thus, the keyframe image retrieval and graph matching tasks can be processed immediately using the existing deep feature matching methods.

3. Methods

3.1. Overview

The proposed method works with the pose graph formulation, where the pose graph can be generated by any keyframe-based visual odometry estimation or VSLAM. Each node of a pose graph is a keyframe consisting of a 6-DoF pose and an encoded image view. The image views can be encoded in different ways for different methods. Our problem formulation is shown in Figure 3; we replaced the feature patch of the image point in DFM with the VGG19 feature maps of the image view. Thus, a pose graph-based local map became a sparse image with the encoded deep features of keyframe image views.

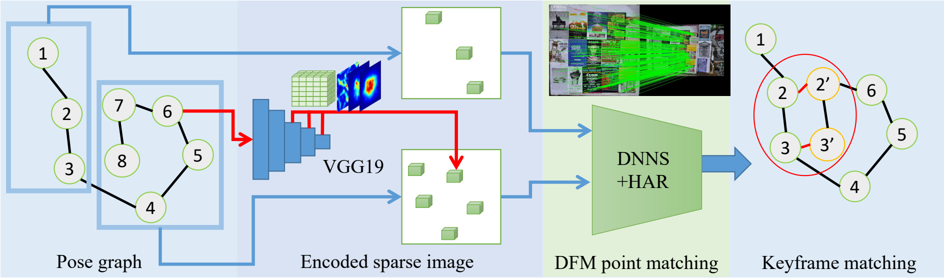

Figure 3. Method overview. The node from 1 to 8 represent the keyframe recorded sequentially.

3.2. Deep Feature Matching

In DFM, the image feature matching process was divided into two stages. In the first stage, the images were warped by estimating a rough geometric transformation between two images in low spatial resolution using DNNS. In the second stage, the DNNS matched the feature patches and refined them hierarchically by HRA, starting from the last layer of a pretrained VGG19 net. Among these processes, the DNNS searched one of the image feature patches to find the best match for each element of the feature patches extracted from another image. We denoted the VGG19 feature maps of the l-th layer $\mathbf{F}^A = VGG19(I^A, l)$ of image A, for a feature point p_i^A in $\mathbf{F}^A$; the DMF found its matched pair $< p^A, p^B >$ in image B by:

$$< p_i^A, p_i^B >= DNNS(\mathbf{F}^A(\Omega(p_i^A)), \mathbf{F}^B(\{\Omega(p_j^B) | j \in neighbors(p_i^A)\})), \quad (1)$$

where the $\Omega(p^A)$ is a point set that represents the receptive field of p^A, and the $neighbors(p^A)$ extracted potential matches near the location of p^A. The best match was defined by the minimum $l2$ distance of the nearest neighbor points. In DFM, the final matching results were refined iteratively using the HRA process, which performed the DNNS matching of all CNN layers. It is clear that the DNNS is the key process for feature point matching.

3.3. Problem Formulation

We first considered the single-agent case. We denoted the k-th node $\mathbf{N}_k$ of a general pose graph as:

$$\mathbf{N}_k = \{\mathbf{T}_k, \mathbf{F}_k\}, \tag{2}$$

where $\mathbf{T}$ represents the 6-DoF transformation from the world frame to the camera frame, and $\mathbf{F}_k$ is the encoded image view of the k-th keyframe.

We first converted the pose graph into a 2D encoded sparse image,

$$\mathbf{p} = \{H(\mathbf{T}_k)|\forall k\}, \tag{3}$$

where $H(\mathbf{T})$ exported the 2D coordinates (x, y) of $\mathbf{T}$ and transferred them into pixel locations. For the descriptors of $\mathbf{p}$, we used the VGG19 net, which contained 5 Conv-layers as the image encoder. Because measurement of the similarities between two images requires a high semantic understanding of the whole image, the feature maps of the first two layers of VGG19 were not considered in our case. The final output was a descriptor that aggregated the feature maps from the 3rd, 4th, and 5th Conv-layers. Thus, for the i-th keyframe, we found its best match in the pose graph by:

$$< p_i, p_j >= DNNS(\mathbf{F}_i, \{\mathbf{F}_j|j \in neighbors(p_i), j \neq i\}), \tag{4}$$

where the feature maps $\mathbf{F}_i$ were obtained by $\mathbf{F}_i = \{VGG19(I_i, l)|l = 3, 4, 5\}$. Hence, the DNNS in our case was equivalent to find the j that minimized the distance between $\mathbf{F}_i$ and $\mathbf{F}_j$:

$$j = \arg\min \|\mathbf{F}_i - \mathbf{F}_j\|. \tag{5}$$

when performing the loop closure detection, the encoded sparse images were constructed from local patches of the same pose graph. It is clear that the proposed method can be used for a multiagent system as long as there are overlaps between the pose graphs.

3.4. Workflow

We did not consider how the loop closure detection process was activated because it varies in different VSLAM algorithms. In this work, we assumed the detection was triggered and two local pose graphs, $\mathbf{G}^A = \{\mathbf{N}_i|i = 1, \ldots, k\}$ and $\mathbf{G}^B = \{\mathbf{N}_j|j = k+1, \ldots, k+m\}$, which represent the old and recent keyframes, respectively, were initialized. This initial condition also suits the multiagent system. We show the workflow of the proposed method using pseudocode.

4. Experiments

We conducted the benchmark evaluation on two self-driving car datasets, i.e., the KITTI and Oxford RobotCar. The method was implemented on Pytorch based on the opensource code of DFM (https://github.com/ufukefe/DFM (accessed on 1 January 2022)) with a pretrained VGG19, and all evaluations were carried out on a desktop PC with a GTX 1080Ti GPU. We first demonstrated the matching performance of the proposed method using the ground truth odometry data of the KITTI sequence 00. The two pose graphs to be matched by the proposed method were generated with random starting and ending frames. We generated a keyframe every 20 m throughout the odometry history. As the ground truth odometry data were used, we identified the good matches by computing their actual location distances. We simply used half the distance between (10 m) two keyframes as the threshold for a mismatch. The matching results are given in Table 1. The results indicate that the proposed method is capable of performing a robust pose graph match over large-scale navigation data when the good matches (inliers) are the majority (>50%) (see Algorithm 1).

Algorithm 1: Deep Pose Graph Matching

```
Data: $\mathbf{G}^A, \mathbf{G}^B$
Result: $\mathbf{M}^{B,A}$
$\mathbf{T}^A, \mathbf{F}^A \leftarrow \mathbf{G}^A$;
$\mathbf{T}^B, \mathbf{F}^B \leftarrow \mathbf{G}^B$;
$\mathbf{p}^A \leftarrow H(\mathbf{T}^A)$;
$\mathbf{p}^B \leftarrow H(\mathbf{T}^B)$;
for $p_i^B$ in $\mathbf{p}^B$ do
    $\mathbf{F}_i \leftarrow \mathbf{F}^B(p_i^B)$;
    $\hat{\mathbf{p}}^A = neighbors(\mathbf{p}^A, p_i^B)$;
    $\hat{\mathbf{F}} \leftarrow \mathbf{F}^A(\hat{\mathbf{p}}^A)$;
    $< p_i^B, p_j^A >= DNNS(\mathbf{F}_i, \hat{\mathbf{F}})$;
    $\mathbf{M}^{B,A}.append(< p_i^B, p_j^A >)$;
end
```

Table 1. Matching results of the pose graphs with random starting and ending frames. The first column shows the total number of keyframes, while the second one counts their overlapping keyframes. We visualize the matching results of random start 1 in Figure 4.

	Total	Overlap	Matches	Good Match (%)	FPS
Random start 1	120:101	62:60	47	55%	8
Random start 2	86:76	43:40	28	64%	14
Random start 3	312:164	36:43	16	72%	3
Random start 4	181:143	53:46	31	58%	8
Random start 5	62:34	8:6	3	66%	15
Average	152:104	40:39	25	63%	10

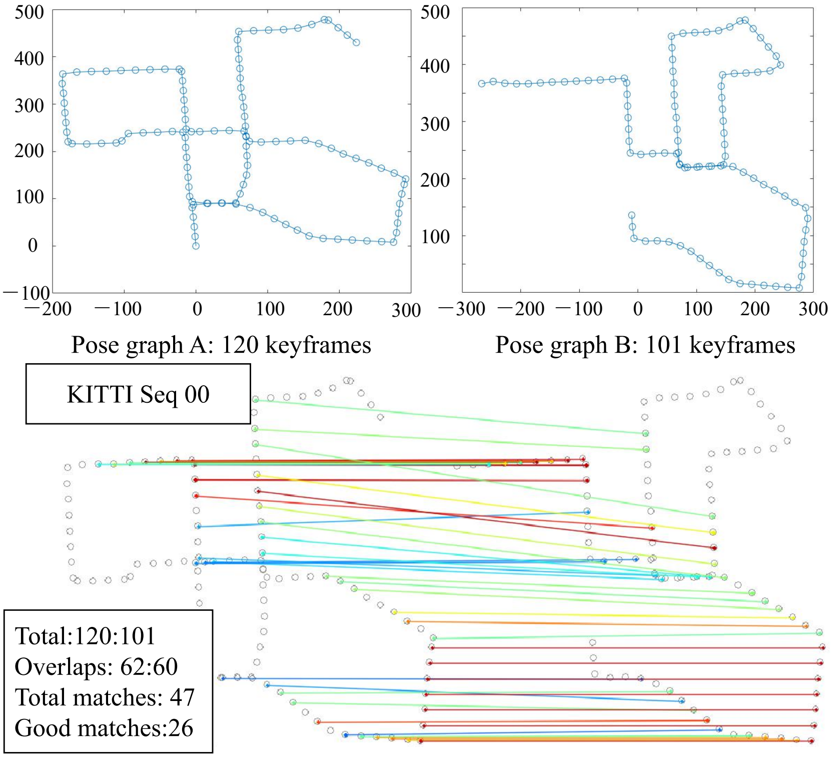

Figure 4. KITTI sequence 00 results.

To further clarify the feasibility and potential of the proposed method's use in real VSLAM and multiagent applications, in the Oxford RobotCar dataset's evaluation, we used the pose graphs generated by our previous visual odometry work [39] during two independent sessions on the global map and a local map, respectively. Then, we performed the same pose graph-matching test, and the result is shown in Figure 5. The matching results showed that the local navigation trajectory was properly matched with the global one. It demonstrates that the proposed method can perform robust keyframe matching for loop closure detection.

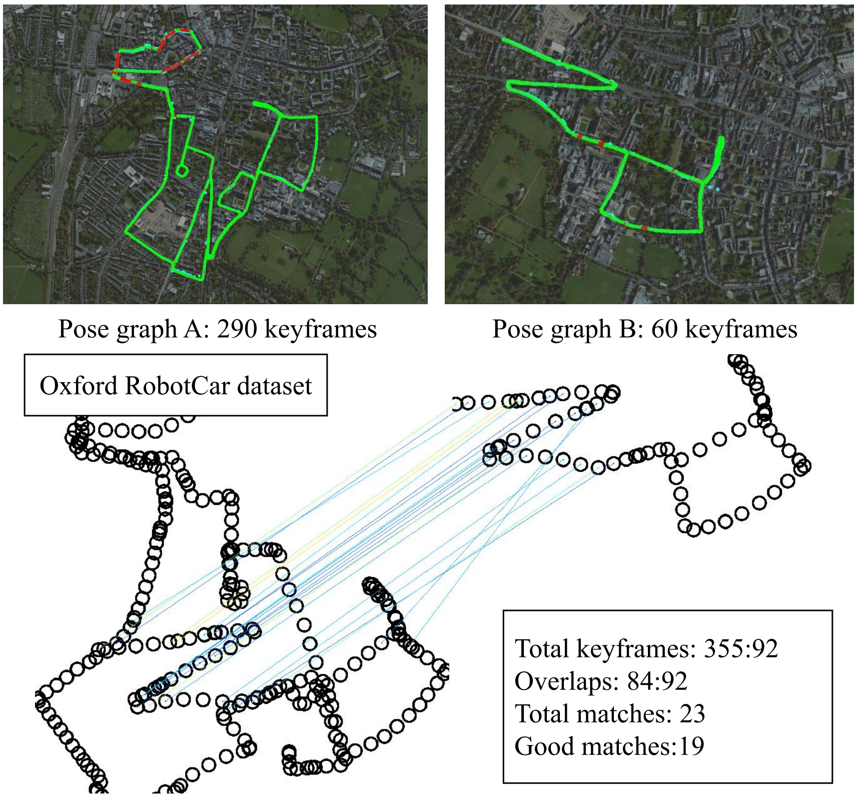

Figure 5. RobotCar results.

Although absolute and relative trajectory errors are widely used for VSLAM or VO algorithm evaluations, many uncontrollable factors, such as pose graph optimization and bundle adjustment (BA), are involved during the trajectory refinement process. To conduct a fair algorithm comparison with controlled factors, we compared our keyframe matching results with HF-Net [54]. The keyframe matching experiments were conducted on RobotCar datasets with random segmented pose graphs. The good match percentages of each test are shown in Table 2. The results indicate that the proposed method had relatively better matching results than HF-Net. This was because the similar keyframes were hard to distinguish for the feature point matching-based methods, such as HF-Net. Our method addressed this problem not only based on a global description of a keyframe but also on a matching results refinement of all similar matches based on geometric consensus. While the HF-Net can match the features between two images in more than 30 FPS, the keyframe matching becomes slow when two large keyframe groups need to be matched. The average FPS of the whole process is given in the last column for both methods.

Table 2. Percentages of good matches using random segmented pose graphs. The best results are marked in bold.

	Test 1	Test 2	Test 3	Test 4	Test 5	Avg. Good Match	Avg. FPS
HF-Net	**67%**	61%	46%	52%	**71%**	59.4%	3.6
Our	59%	**64%**	**73%**	**58%**	68%	64.4%	6.1

To compare the final loop closure detection performance quantitatively, we performed standard pose graph optimization in the g2o toolbox across the above matching results. The translation error of the relative pose error every 50 m after the loop closure correction is given in Figure 6.

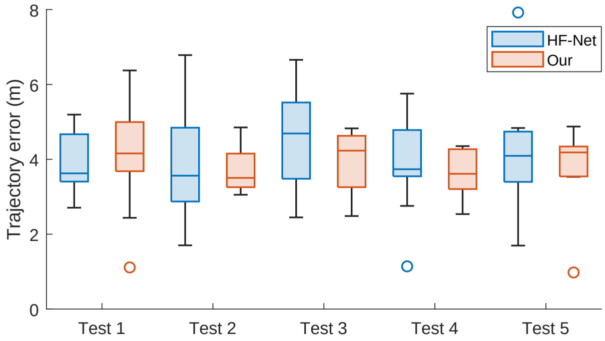

Figure 6. The translation error of relative pose error (every 50 m) after loop closure correction based on the matching results of Table 2.

To validate the potential for applying the proposed method to multiagent cases, we conducted the flight navigation test with two UAVs as Figure 7 shows. In this setup, each UAV was flying based on its own coordinate system. When the proposed method was activated, and their keyframes were matched, their pose graphs were transformed into a unified and global coordinate system, which offered the precondition for multiagent control.

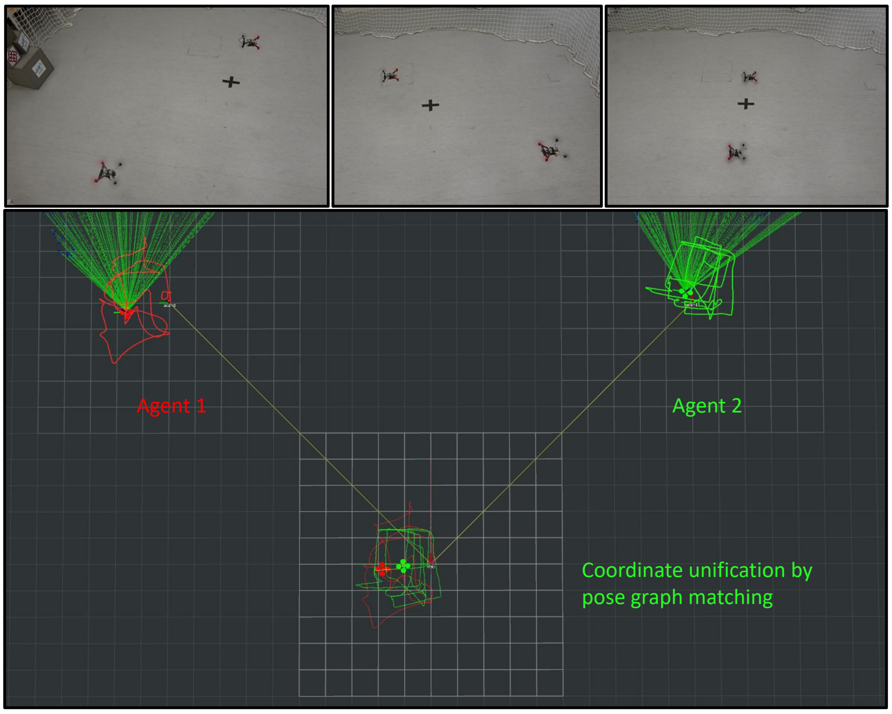

Figure 7. Validation on multiagent application.

5. Conclusions

This work demonstrates that the deep feature matching approach can solve the pose graph matching problem by considering a whole pose graph as an image and replacing image point descriptors with keyframe descriptors. The proposed method was built on the deep learning-based feature point matching method DFM. Instead of the receptive field of the feature point, we used the high-level semantic feature maps of the whole keyframe image to perform the DNNS matching. Thus, the overlapped keyframes of the two pose graphs were matched with a similar process for feature point matching. The proposed method was evaluated on two well-known self-driving car benchmark datasets. The experimental results have shown the feasibility of and potential for employing the proposed method in semantic VSLAM loop closure detection and multiagent navigation. In the future, we will transfer the proposed method to another deep point matching method called SuperGlue, because in SuperGlue, a graph neural network (GNN) is used to assist the matching process, making use of the connection relationship between points.

Author Contributions: Conceptualization, R.D.; Data curation, R.D. and Y.F.; Formal analysis, R.D.; Funding acquisition, C.-Y.W.; Investigation, R.D. and Y.F.; Methodology, R.D.; Project administration, C.-Y.W.; Resources, R.D.; Software, R.D.; Validation, R.D.; Visualization, R.D.; Writing—original draft, R.D.; Writing—review & editing, R.D. and Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the PolyU Start-up Fund (P0034164, P0036092) and the Research Centre for Unmanned Autonomous Systems (P0042699).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was funded by the PolyU Start-up Fund (P0034164, P0036092) and the Research Centre for Unmanned Autonomous Systems (P0042699).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Davison, A.J. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; p. 1403.
2. Li, Y.; Zhang, H.; Liang, X.; Huang, B. Event-Triggered-Based Distributed Cooperative Energy Management for Multienergy Systems. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2008–2022. [CrossRef]
3. Li, T.; Yang, D.; Xie, X.; Zhang, H. Event-Triggered Control of Nonlinear Discrete-Time System With Unknown Dynamics Based on HDP(λ). *IEEE Trans. Cybern.* **2022**, *52*, 6046–6058. [CrossRef] [PubMed]
4. Zhang, N.; Sun, Q.; Yang, L.; Li, Y. Event-Triggered Distributed Hybrid Control Scheme for the Integrated Energy System. *IEEE Trans. Ind. Inform.* **2022**, *18*, 835–846. [CrossRef]
5. Latif, Y.; Cadena, C.; Neira, J. Robust loop closing over time for pose graph SLAM. *Int. J. Robot. Res.* **2013**, *32*, 1611–1626. [CrossRef]
6. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
7. Feng, Y.; Tse, K.; Chen, S.; Wen, C.Y.; Li, B. Learning-based autonomous uav system for electrical and mechanical (E&m) device inspection. *Sensors* **2021**, *21*, 1385. [PubMed]
8. Chang, C.W.; Lo, L.Y.; Cheung, H.C.; Feng, Y.; Yang, A.S.; Wen, C.Y.; Zhou, W. Proactive Guidance for Accurate UAV Landing on a Dynamic Platform: A Visual–Inertial Approach. *Sensors* **2022**, *22*, 404. [CrossRef]
9. Jiang, B.; Li, B.; Zhou, W.; Lo, L.Y.; Chen, C.K.; Wen, C.Y. Neural Network Based Model Predictive Control for a Quadrotor UAV. *Aerospace* **2022**, *9*, 460. [CrossRef]
10. Dai, X.; Long, S.; Zhang, Z.; Gong, D. Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Front. Neurorobot.* **2019**, *13*, 15. [CrossRef]
11. Stachniss, C.; Leonard, J.J.; Thrun, S. Simultaneous localization and mapping. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1153–1176.
12. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [CrossRef]
13. Scaramuzza, D.; Fraundorfer, F. Visual odometry [tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [CrossRef]

14. Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. *Nature* **1981**, *293*, 133–135. [CrossRef]
15. Harris, C.G.; Pike, J. 3D positional integration from image sequences. *Image Vis. Comput.* **1988**, *6*, 87–90. [CrossRef]
16. Duan, R.; Fu, C.; Kayacan, E. Tracking–recommendation–detection: A novel online target modeling for visual tracking. *Eng. Appl. Artif. Intell.* **2017**, *64*, 128–139. [CrossRef]
17. Karmokar, P.; Dhal, K.; Beksi, W.J.; Chakravarthy, A. Vision-Based Guidance for Tracking Dynamic Objects. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 1106–1115. [CrossRef]
18. Dhal, K.; Karmokar, P.; Chakravarthy, A.; Beksi, W.J. Vision-Based Guidance for Tracking Multiple Dynamic Objects. *J. Intell. Robot. Syst.* **2022**, *105*, 66. [CrossRef]
19. Huang, Z.; Fu, C.; Li, Y.; Lin, F.; Lu, P. Learning Aberrance Repressed Correlation Filters for Real-Time UAV Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
20. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Lu, G. AutoTrack: Towards High-Performance Visual Tracking for UAV with Automatic Spatio-Temporal Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Seattle, WA, USA, 13–19 June 2020.
21. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. HiFT: Hierarchical Feature Transformer for Aerial Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021.
22. Grisetti, G.; Kümmerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. [CrossRef]
23. Bednář, J.; Petrlík, M.; Vivaldini, K.C.T.; Saska, M. Deployment of Reliable Visual Inertial Odometry Approaches for Unmanned Aerial Vehicles in Real-world Environment. In Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, 21–24 June 2022; pp. 167–176. [CrossRef]
24. Mulmuley, K. Computational geometry. In *An Introduction through Randomized Algorithms*; Prentice-Hall: Hoboken, NJ, USA, 1994.
25. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
26. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
27. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.
28. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
29. Strasdat, H.; Davison, A.J.; Montiel, J.M.; Konolige, K. Double window optimisation for constant time visual SLAM. In Proceedings of the 2011 International Conference on Computer Vision, Tokyo, Japan, 25–27 May 2011; pp. 2352–2359.
30. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-based visual-inertial slam using nonlinear optimization. In Proceedings of the Robotis Science and Systems (RSS) 2013, Berlin, Germany, 24–28 June 2013.
31. Jiang, C.; Paudel, D.P.; Fougerolle, Y.; Fofi, D.; Demonceaux, C. Static-Map and Dynamic Object Reconstruction in Outdoor Scenes Using 3-D Motion Segmentation. *IEEE Robot. Autom. Lett.* **2016**, *1*, 324–331. [CrossRef]
32. Chen, S.; Wen, C.Y.; Zou, Y.; Chen, W. Stereo Visual Inertial Pose Estimation Based on Feedforward-Feedback Loops. *arXiv* **2020**, arXiv:2007.02250.
33. Chen, S.; Zhou, W.; Yang, A.S.; Chen, H.; Li, B.; Wen, C.Y. An End-to-End UAV Simulation Platform for Visual SLAM and Navigation. *Aerospace* **2022**, *9*, 48. [CrossRef]
34. Li, X.; Ling, H. PoGO-Net: Pose Graph Optimization with Graph Neural Networks. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 5875–5885. [CrossRef]
35. Yang, T.; George, J.; Qin, J.; Yi, X.; Wu, J. Distributed least squares solver for network linear equations. *Automatica* **2020**, *113*, 108798. [CrossRef]
36. Li, Y.; Gao, D.W.; Gao, W.; Zhang, H.; Zhou, J. Double-Mode Energy Management for Multi-Energy System via Distributed Dynamic Event-Triggered Newton-Raphson Algorithm. *IEEE Trans. Smart Grid* **2020**, *11*, 5339–5356. [CrossRef]
37. Twinanda, A.P.; Meilland, M.; Sidibé, D.; Comport, A.I. On Keyframe Positioning for Pose Graphs Applied to Visual SLAM. Available online: https://hal.archives-ouvertes.fr/hal-01357358/document (accessed on 1 January 2021).
38. Li, Y.; Gao, D.W.; Gao, W.; Zhang, H.; Zhou, J. A Distributed Double-Newton Descent Algorithm for Cooperative Energy Management of Multiple Energy Bodies in Energy Internet. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5993–6003. [CrossRef]
39. Duan, R.; Paudel, D.P.; Fu, C.; Lu, P. Stereo Orientation Prior for UAV Robust and Accurate Visual Odometry. *IEEE/ASME Trans. Mechatronics* **2022**, 1–11. [CrossRef]
40. Li, Y.; Wang, J.; Wang, R.; Gao, D.W.; Sun, Q.; Zhang, H. A Switched Newton-Raphson-Based Distributed Energy Management Algorithm for Multienergy System Under Persistent DoS Attacks. *IEEE Trans. Autom. Sci. Eng.* **2021**, 1–13. [CrossRef]
41. Li, Y.; Li, T.; Zhang, H.; Xie, X.; Sun, Q. Distributed Resilient Double-Gradient-Descent Based Energy Management Strategy for Multi-Energy System Under DoS Attacks. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2301–2316. [CrossRef]
42. Strasdat, H.; Montiel, J.; Davison, A.J. Scale drift-aware large scale monocular SLAM. *Robot. Sci. Syst. VI* **2010**, *2*, 7.
43. Efe, U.; Ince, K.G.; Alatan, A. DFM: A Performance Baseline for Deep Feature Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nashville, TN, USA, 20–25 June 2021; pp. 4284–4293.

44. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
45. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 Year, 1000km: The Oxford RobotCar Dataset. *The Int. J. Robot. Res. IJRR* **2017**, *36*, 3–15. [CrossRef]
46. Kejriwal, N.; Kumar, S.; Shibata, T. High performance loop closure detection using bag of word pairs. *Robot. Auton. Syst.* **2016**, *77*, 55–65. [CrossRef]
47. Duan, R.; Fu, C.; Kayacan, E. Recoverable recommended keypoint-aware visual tracking using coupled-layer appearance modelling. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4085–4091. [CrossRef]
48. Yue, Y.; Zhao, C.; Wu, Z.; Yang, C.; Wang, Y.; Wang, D. Collaborative Semantic Understanding and Mapping Framework for Autonomous Systems. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 978–989. [CrossRef]
49. Yue, Y.; Zhao, C.; Li, R.; Yang, C.; Zhang, J.; Wen, M.; Wang, Y.; Wang, D. A Hierarchical Framework for Collaborative Probabilistic Semantic Mapping. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9659–9665. [CrossRef]
50. Yang, L.; Sun, Q.; Zhang, N.; Li, Y. Indirect Multi-Energy Transactions of Energy Internet with Deep Reinforcement Learning Approach. *IEEE Trans. Power Syst.* **2022**, *37*, 4067–4077. [CrossRef]
51. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, Seattle, WA, USA, 13–19 June 2020.
52. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 224–236.
53. Duan, R.; Fu, C.; Alexis, K.; Kayacan, E. Online Recommendation-based Convolutional Features for Scale-Aware Visual Tracking. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14206–14212. [CrossRef]
54. Sarlin, P.E.; Cadena, C.; Siegwart, R.; Dymczyk, M. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.