2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

AutoFusion: Autonomous Visual Geolocation and Online Dense Reconstruction for UAV Cluster

Yizhu Zhang[1], Shuhui Bu[1]*, Yifei Dong[1], Yu Zhang[1], Kun Li[1], Lin Chen[1]

***Abstract*— Real-time dense reconstruction using Unmanned Aerial Vehicle (UAV) is becoming increasingly popular in large-scale rescue and environmental monitoring tasks. However, due to the energy constraints of a single UAV, the efficiency can be greatly improved through the collaboration of multi-UAVs. Nevertheless, when faced with unknown environments or the loss of Global Navigation Satellite System (GNSS) signal, most multi-UAV SLAM systems can't work, making it hard to construct a global consistent map. In this paper, we propose a real-time dense reconstruction system called *AutoFusion* for multiple UAVs, which robustly supports scenarios with lost global positioning and weak co-visibility. A method for Visual Geolocation and Matching Network (VGMN) is suggested by constructing a graph convolutional neural network as a feature extractor. It can acquire geographical location information solely through images. We also present a real-time dense reconstruction framework for multi-UAV with autonomous visual geolocation. UAV agents send images and relative positions to the ground server, which processes the data using VGMN for multi-agent geolocation optimization, including initialization, pose graph optimization, and map fusion. Extensive experiments demonstrate that our system can efficiently and stably construct large-scale dense maps in real-time with high accuracy and robustness.**

I. Introduction

In the fields of geological exploration, urban planning, cultural heritage preservation, environmental monitoring, and more, the use of Unmanned Aerial Vehicle equipped with sensors for dense reconstruction [1], [2] is an important technology. This technology utilizes aerial imagery data collected by UAVs and employs image processing and computer vision techniques to achieve high-precision 3D reconstruction of targets such as the land's surface or buildings. In dense reconstruction, to maintain consistency with actual terrain and structures, it is necessary to generate high-precision point cloud data [3]–[5] and continuous three-dimensional models. Traditional 3D reconstruction methods [6], [7] still rely on Structure from Motion (SfM), but they function in an offline capacity and demand significant time consumption, making them unsuitable for rapid and real-time requirements. Simultaneous Localization and Mapping (SLAM) technology can perform pose estimation and realize incremental dense reconstruction. However, the battery capacity and flight speed limitations of single UAV restrict the scope of terrain reconstruction. Therefore, for large-scale dense reconstruction tasks, the collaborative operation of multiple UAVs [8] can efficiently complete the mission and enhance performance.

The foundation of multi-UAV collaborative reconstruction is that each UAV operates its own SLAM system [9], but a unified fused map requires high-precision pose information between the aerial images captured by different UAVs. Most current multi-UAV SLAM frameworks [10] are limited to indoor or relatively small outdoor scenes, and they struggle to construct maps effectively in large-scale high-altitude scenarios. These frameworks require the relative transformation of UAVs to complete map fusion [11]–[13], but when UAV loses its global geographical location or there is low image similarity between UAVs, the systems can't work.

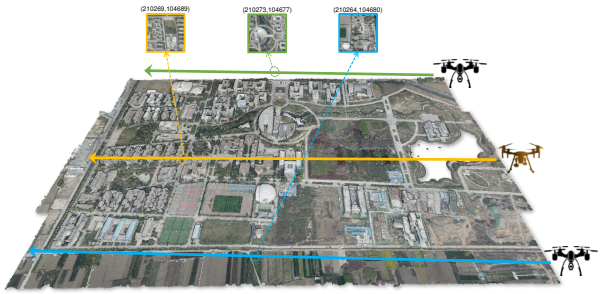


Fig. 1. A dense reconstruction result for UAVs. The green, yellow, and blue lines represent the flight paths. Circular markers indicate keyframes. These markers point to the corresponding tile with geolocation information retrieved through Visual Geolocation and Matching Network.

To address these issues, in this paper we first propose a visual geolocation and matching network, which adapts to large scales and reflects the relationships between the local features of the image. It employs a graph convolutional neural network as a feature extractor to obtain global geolocation information from aerial images, and serves as an image retrieval and matching method to rank co-visible images between UAVs. A framework for real-time dense reconstruction with multiple UAVs is constructed, where each UAV operates its own SLAM system as an agent and sends its image information to the ground server. The server initializes the relative transformation between UAVs by establishing a local optimization with weighted geolocation information constraints. This is a self-adjusting optimization method that adapts itself based on the quality of image similarity, allowing initialization even without any co-visibility between UAVs. Geolocation information is also used in conjunction with SLAM's relative poses for backend pose graph optimization. Experimental results demonstrate that our method effectively improves the robustness of dense reconstruction in unknown scenes and the accuracy of localization. The main contributions of this work are as follows:

- a visual geolocation and matching network method,

*Corresponding author (bushuhui@nwpu.edu.cn)
[1]School of Aeronautics, Northwestern Polytechnical University

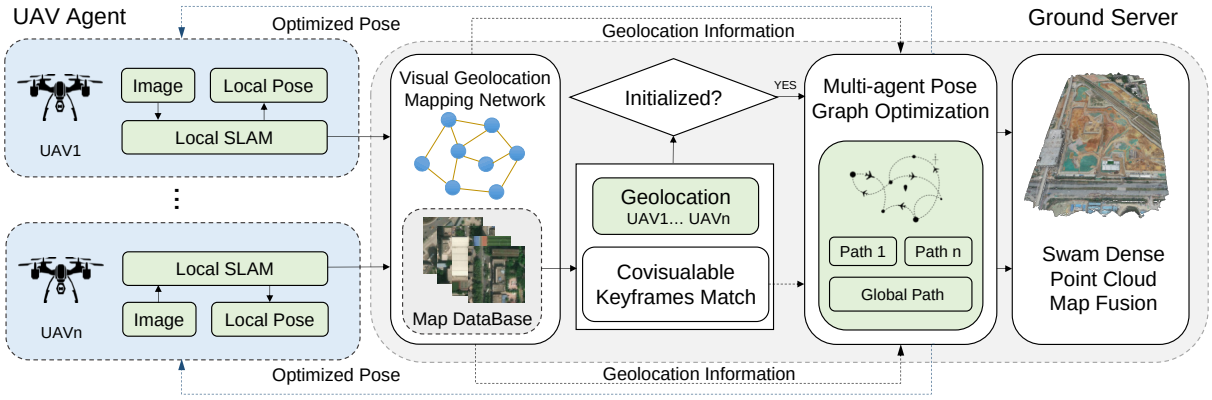


Fig. 2. Overview of the system. The multi-UAV dense reconstruction system consists of UAV agent and a ground server. UAV agents run SLAM system. The server runs the visual geolocation and matching network, initialization, and pose optimization.

which can acquire UAV's geolocation and complete co-visibility matching between similar images,

- a system for multi-UAV collaborative dense reconstruction, in which each UAV operates its SLAM, and the ground server performs multi-agent weighted geolocation optimization and map fusion, and
- three large-scale aerial dataset collections are designed to effectively evaluate multi-UAV real-time dense reconstruction.

II. Related Works

In this section, we review the relevant methods for multi-agent SLAM dense reconstruction and deep learning-based geolocation.

A. Multi-UAV SLAM System

With the rapid development of UAV and SLAM technology, researchers have turned their attention to multi-UAV dense SLAM techniques, considering the limitations of single UAV. CVI-SLAM [14] provides a centralized collaborative SLAM system that offloads resource-intensive tasks to a central server, which shares information with the UAVs. CCM-SLAM [9] introduces a centralized collaborative monocular visual SLAM system, consisting of client devices with computational capabilities and a ground server. Clients are responsible for the front-end, while the server receives data from clients for global pose optimization and map fusion. COVINS [15] reevaluates key components of centralized collaborative SLAM, emphasizing improvements in accuracy and scalability. COVINS-G [16] builds upon this, modularizing the back-end system and proposing a generic visual SLAM back-end component compatible with various Visual-Inertial Odometry (VIO) [17], [18] front-ends. Existing collaborative methods mainly focus on improving localization performance and pay little attention to dense reconstruction tasks in large-scale aerial surveys.

Regarding dense reconstruction, traditional Structure from Motion methods are limited by their algorithmic framework, preventing real-time completion of reconstruction tasks. CoScan [19] achieves dense reconstruction of unknown indoor environments through collaborative scanning. Kimera-Multi [20] extends the Kimera [21] framework to develop a distributed multi-robot metric-semantic SLAM system, optimizing distributed pose graphs and adjusting local deformation grids. Coxgraph [22] focuses on transmitting submaps in multi-robot systems using a compact network packet to perform SDF reconstruction [23] at the terminal. However, these methods generally require sufficient co-visibility [24] between UAVs to create a Bag of Word (BoW) [25] for matching, which is hard to apply to real-time reconstruction robustly in large-scale scenarios.

B. Visual Geolocation

Geolocation [26] through visual information has garnered attention, and methods based on VLAD [27] have been widely used. The introduction of NetVLAD [28] makes the entire process learnable and implements it as a closed-loop neural network. It uses a Convolutional Neural Network (CNN) [29] as an image feature extractor and flexibly employs CNN layers for learning the entire process. Patch-NetVLAD [30] builds upon NetVLAD by applying accelerated multi-scale patch features to describe VLAD cluster features. These methods all employ contrastive learning [31] as their training framework, where similar data points are brought close to each other in the learned space. Another category of visual geolocation methods treats it as a classification [32] problem. The idea behind such methods is that two images from the same region may share the same semantic information, even if the scenes are different. Gabriele [33] points out that an approach using classification is possible, but it cannot actually solve the problem. We propose a method that uses graph neural network as the foundation for feature extraction network and constructs a training framework for image classification to meet geolocation requirements.

III. Methodology

The system for multi-UAV dense mapping is depicted in Figure 2. The overall framework is divided into two parts: the

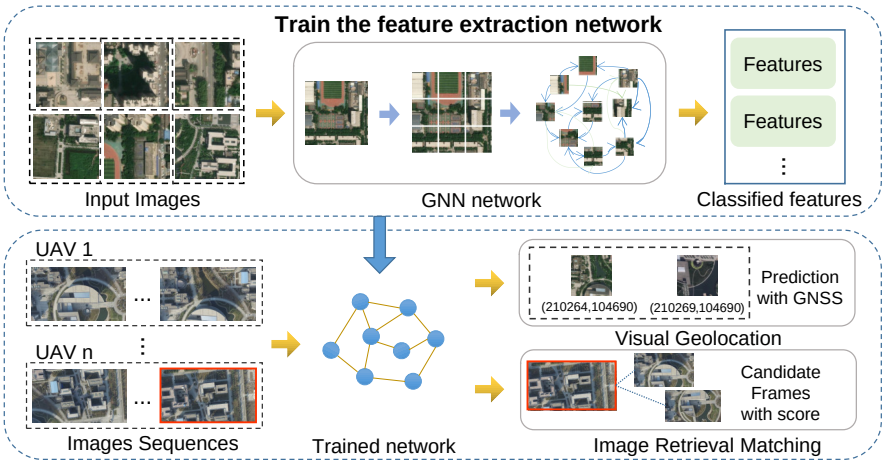


Fig. 3. The above box shows the pipeline for the Visual Geolocation and Matching Network, which uses a graph convolutional network to extract features. The box below illustrates the usage of geolocation and matching.

UAV agent and the ground server. Each UAV independently runs its own SLAM system and sends keyframe information to the ground server. VGMN receives images from UAVs to perform similarity matching and acquire their geolocation information by retrieving them in the dataset. Then, using a multi-agent weighted bundle adjustment (BA) optimization, we establish the initial relative transformation. Geolocation information and relative pose are jointly input into the optimization, finally providing accurate pose output for map fusion.

A. Visual Geolocation and Matching Network

The entire process involves constructing dataset, training the network, and geolocation with matching. A feature extraction method is presented, which is based on image classification and Graph Neural Network (GNN). The structure of the graph can represent the feature relationships within the image, which enhances the accuracy of classification. The geographical coordinates of each image serve as labels, and the images are grouped such that the distance between images within each group does not fall below a specified threshold. The GNN outputs features are used for image similarity matching between UAVs and retrieving corresponding geolocation information from the database. The specific workflow is depicted in Figure 3.

1) Dataset: A portion of the dataset we use is derived from remote sensing maps, where each image includes Mercator (UTM) coordinate [34] labels containing longitude and latitude information. The dataset also includes satellite imagery maps of the respective regions, which are divided into multiple areas based on the map's tile levels. The UTM coordinates of each area are used as corresponding labels. Each tile can be converted into specific longitude and latitude information based on its hierarchical level, denoted as z, using the following conversion:

$$\begin{cases} lon = \frac{x}{2^z} \cdot 360 - 180, \\ lat = arctan(sinh(\pi - \frac{y}{2^z} \cdot 2\pi)) \cdot \frac{180}{\pi}, \end{cases} \tag{1}$$

where x and y represent the tile numbers, and z indicates the hierarchical level of the tiles, determined based on different shooting altitudes. To prevent the influence of similar information from neighboring tiles, we group the satellite images with coordinate labels. We ensure that the images within each group are separated by a distance not less than the specified inter-group interval. We select 9 images within each group, with an interval of 3 between each group.

2) Feature Extraction Network: For each image, we employ a Graph Convolutional Network (GCN) to extract features. Initially, we utilize a CNN to extract N-dimensional features from the image. The image is divided into M patches, creating an $M \times N$-dimensional feature matrix. Each of the M patches is treated as a node v, and we construct a set $V = \{v_1, v_2, ..., v_n\}$ from these nodes. We connect each node to its neighboring node, forming an edge $e_{ij} \in E$. The feature vector for each node is denoted as h_i and our objective is to update the features of each node in a manner that allows them to continuously propagate and integrate information from their neighboring nodes.

$$h_i' = ReLU\left(\sum_{j \in \mathbf{N}(i)} \frac{\boldsymbol{W} \cdot h_j}{\sigma}\right), \tag{2}$$

$$\sigma = \sqrt{|\mathbf{N}(i)| \cdot |\mathbf{N}(j)|}, \tag{3}$$

where h_i represents the input feature of node v_i, h_i' is the updated output feature of node v_i. The parameter $\mathbf{N}(i)$ denotes the set of nodes adjacent to node v_i. $|\mathbf{N}(i)|$ indicates the number of neighboring nodes for node v_i. $\boldsymbol{W}$ is the parameter matrix used for linear transformation of node features, and $ReLU()$ is employed as the activation function. The updated feature h_i' of node v_i is obtained by linearly transforming and weighted summing the features h_j of all neighboring nodes v_j. The parameter σ controls the weight to ensure that it is not influenced by the number of node connections, enhancing stability in this way. Final network includes the core GCN described above, along with a pooling layer and a fully connected layer, to better adapt to the

ultimate loss function.

3) Geolocation and Matching: Using a trained feature extraction network, we can obtain latitude and longitude information for any input UAV image. However, the tile-level resolution often cannot match the resolution of the images captured by UAVs, resulting in some degree of error. We divide the original image into blocks of tile resolution, input all these blocks into the network for localization. For the image blocks that conform to the continuity of tile labels, we find their center , and based on the offset of center in the original image, calculate more accurate GNSS information. In the matching task, each image yields descriptor information through the feature extractor. We calculate distances between descriptors and assign scores, and then rank adjacent UAV frames based on these scores to obtain the best co-visible frames.

B. Multi-UAV Dense SLAM

In multi-UAV dense SLAM system, each UAV agent has its own monocular Visual Odometry (VO). The server receives images and handles the remaining tasks, including initialization, pose optimization, and map fusion. To merge dense maps from multiple UAVs, it is necessary to initialize coordinate transformations among the UAVs. Below, we introduce the specific methods for each module.

1) Initialization: To obtain relative transformations among the UAVs, we use the VGMN to extract features from images during the initialization process, resulting in descriptor information. Next, construct a list of similar images for each keyframe and select the images with higher ratings. VGMN also obtains geolocation information from images. We construct a multi-agent weight bundle adjustment optimization to compute relative transformaion of UAVs. The optimization objective is as follows:

$$\min \sum_{j=1}^{V} \sum_{i=1}^{U} ||\boldsymbol{P}_{w,0}^{i} - (\boldsymbol{K} \cdot \boldsymbol{R}_{w}^{c} \cdot \boldsymbol{P}_{c,j}^{i} + \boldsymbol{t}_{w}^{c})||^2 + w_{geo} \cdot ||\boldsymbol{t}_{w}^{c} - s \cdot \boldsymbol{t}_{geo}||^2, \tag{4}$$

where $\boldsymbol{R}_{w}^{c}$ represents the rotation transformation from the coordinate system of the followed UAV(c) to the central UAV(w). $\boldsymbol{t}_{w}^{c}$ denotes the translation transformation. $\boldsymbol{K}$ represents the camera intrinsic parameter matrix. We select the first frame of the central UAV(w) with the i-th observed point, and $\boldsymbol{P}_{c,j}^{i}$ denotes the i-th observed point in the j-th frame in the coordinate system c. U and V respectively represent the number of observation points and the number of keyframes. The weight coefficient for geolocation information in the optimization is denoted as w_{geo} and is adjusted based on the quality of co-visibility. When co-visibility is completely lost, only geolocation information is used for optimizing the translation transformation. During this process, it's necessary to align the scale information s and use the Haversine formula for this purpose:

$$s = \frac{\sqrt{(t_x^k - t_x^0)^2 + (t_y^k - t_y^0)^2}}{\lambda \cdot D}, \tag{5}$$

$$D = arccos\big[\,\big(sin(G_{lat}^k)sin(G_{lat}^0)\big) + cos(G_{lat}^k)cos(G_{lat}^0)cos(G_{lon}^k - G_{lon}^0)\big], \tag{6}$$

where G_{lat}^k and G_{lon}^k represent the latitude and longitude information of the k-th frame, while t_x^k and t_y^k denote the x and y coordinates in the VO coordinate system. The constant λ is set to 3.963, resulting in a final scale unit of meters (m).

2) Pose Graph Optimization: After the initialization, the poses of the followed UAV's keyframes in center coordinate system can be calculated using $\boldsymbol{R}_{w}^{q} = \boldsymbol{R}_{w}^{c}\boldsymbol{R}_{c}^{q}$ and $\boldsymbol{t}_{w}^{q} = \boldsymbol{R}_{w}^{c}\boldsymbol{t}_{c}^{q} + \boldsymbol{t}_{w}^{c}$. The parameter $\boldsymbol{R}_{w}^{q}$ and $\boldsymbol{t}_{w}^{q}$ represent the rotation and translation of the followed UAV's q-th keyframe in the central coordinate system. $\boldsymbol{R}_{c}^{q}$ and $\boldsymbol{t}_{c}^{q}$ are in the NED (North-East-Down) coordinate system of the followed UAVs. To eliminate cumulative errors in pose estimation, a sliding window pose optimization [35] method is used. Geolocation information is used as a constraint in the pose optimization [36]. The objective we optimize is:

$$\min \Big\{ \sum_{q=1}^{n} \Big(\sum_{s=1}^{m} ||\boldsymbol{e}_{q,q+s}||^2_{\boldsymbol{\Sigma}_{q,q+s}} + \sum_{g \in \mathcal{G}} ||\boldsymbol{e}_{q,q+g}||^2_{\boldsymbol{\Sigma}_{q,q+g}} \Big) + \rho\Big(\sum_{i,j \in \mathcal{C}} ||\boldsymbol{e}_{i,j}||^2_{\boldsymbol{\Sigma}_{i,j}} \Big) \Big\}, \tag{7}$$

where $\boldsymbol{e}_{i,j}$ represents the standard relative pose residual between the i-th and j-th frames, $||\boldsymbol{e}_{ij}||^2_{\boldsymbol{\Sigma}_{ij}} = \boldsymbol{e}_{ij}^T \boldsymbol{\Sigma}_{ij} \boldsymbol{e}_{ij}$ represents the Mahalanobis distance of the covariance matrix $\boldsymbol{\Sigma}$. In the first term, it denotes the residual between the current UAV's q-th frame and the s-th frame within its sliding window (where the sliding window size is m, ultimately set to $m = 6$). $\mathcal{G}$ is a set of geolocation information, and $g = \{s | s \in \mathcal{G}\}$ indicates frames within the sliding window that contain geolocation. $\mathcal{C}$ includes edges between all co-visual frames in the central UAV, where ρ is the robust Huber loss function. The optimization algorithm we employ is the Levenberg-Marquardt.

3) Dense Mapping and Fusion: To obtain real-time dense map for each UAV, we reference the method used in DenseFusion [37]. This involved optimizing poses, constructing stereo pairs from images at different time intervals, and using depth maps to generate dense maps. In the process of constructing dense disparity maps, a fast stereo matching method called ELAS [38] is employed. The server generate dense reconstruction for each UAV's submap and unify the map points to the central coordinate system through relative transformations. To handle overlapping regions between submaps, we use voxel occupancy to eliminate redundancies. We index the point cloud chunks based on their location information, ensuring that overlapping point clouds are only generated once in the fused map. This approach effectively save memory space and reduce the size of the dense point cloud in the fused map.

IV. Experiments

In this section, we evaluate the performance of VGMN and the multi-UAV dense SLAM by constructing datasets. VGMN is implemented using Python, and the network training is conducted on a PC with two GeForce GTX 3090

GPUs. The SLAM algorithm is implemented in C++, and the server runs on a PC with an Intel Core i5-12490F CPU, GeForce GTX 3060 GPU, and 16GB of RAM. The experiment is conducted by playing back pre-recorded dataset captured by UAVs in real-time. The agent a NUC 11PAH embedded computer with an Intel Core i7-1165G7 CPU. The server and the agents are connected via a real wireless network to enable real-time communication. Through multiple runs, this ensures both persuasiveness and stability of results.

Fig. 4. The 18-level tiles of the Google satellite map in *npuchangan*. The numbers represent tile information.

A. Visual Geolocation and Matching Network

The dataset consists of satellite and aerial images. A total of 52800 satellite images obtained from Google Maps at a tile level of 18 for the 10752 km^2 city. Some tiles are as shown in the Figure 4. Additionally, there are 800 aerial images from a publicly available UAV dataset that includes GNSS information. To simulate real-world interference conditions, we apply cropping, rotation, and occlusion to the images as supplements to the test set. We evaluate the network's computational time on different resolution input images on the GeForce GTX 3060 computer. The results in Table I indicate that the performance is entirely acceptable for large-scale reconstruction tasks.

TABLE I

VGMN RETRIEVAL OF EACH IMAGE AT DIFFERENT RESOLUTIONS COMPUTATIONAL TIME. TIME IS PROVIDED IN SECONDS(S).

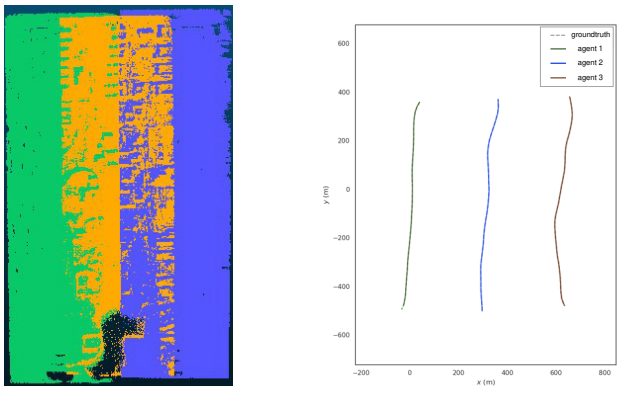
Size	**1920×1080**	**4000×3000**	**5472×3648**
time (s)	0.031	0.044	0.048

TABLE II

RECALL RATES ACROSS DIFFERENT DATASETS.

Dataset	***npuyouyi***	***npuchangan***	***famensi***	***wholecity***
NetVLAD (R@1)	53.7	57.9	62.8	51.2
Ours (R@1)	**75.6**	**79.4**	**83.5**	**73.5**
NetVLAD (R@5)	61.4	62.6	67.5	54.8
Ours (R@5)	**84.2**	**86.7**	**89.2**	**79.4**

We also construct three large-scale datasets for areas of 3±0.2 km^2. The accuracy rates of matching and retrieval are above 95%, but recall is a better indicator of whether we can successfully identify the true location. We validate the accuracy of geolocation by evaluating the recall rate as Table II shows. Compared to NetVLAD, our network is better suited for large-scale scenarios, fully leveraging the data, and effectively capturing the relationships between local image features. The size of the area has a direct impact on the results, as large-scale area contains similar features such as grasslands and water which are difficult to distinguish.



(a) (b)

Fig. 5. (a) The dense point clouds colored by agent's id. *Green*: agent 1. *Yellow*: agent 2. *Blue*: agent 3. (not yet fused). (b) The green, blue, and yellow lines represent the trajectories of three agents. The dashed line represents the ground truth's trajectory (x-y axes).

B. Dense Restruction

The dataset is obtained from three places: *npuyouyi*, *npuchangan* and *famensi*. Each place provides images with Real-Time Kinematic (RTK) information, and the images have a resolution of 4000×3000 pixels. To obtain a high-precision dense point cloud map, high image resolution is required, therefore communication protocols are well designed for reducing duplicated data transfer. We perform offline dense reconstruction using COLMAP [7] and real-time reconstruction using DenseFusion for comparison. To evaluate multi-UAV dense SLAM system, we run three agents separately. The dense point cloud result is as shown in Figure 6. We compare the system's computational time, and Table III shows that the COLMAP consumes a significant amount of time. In contrast, when compared to the single-UAV method, our multi-UAV reconstruction effectively improves system performance.

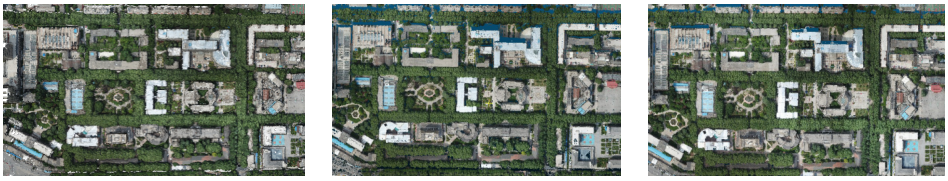
TABLE III

COMPUTATIONAL TIME IN COLMAP, DENSEFUSION, AND OURS. TIME IS PROVIDED IN SECONDS(S).

Dataset	***npuyouyi***	***npuchangan***	***famensi***
COLMAP	1011	1286	1269
DenseFusion	82	86	91
Ours	**31**	**33**	**37**

We evaluate the Absolute Trajectory Error (ATE) and the error in the dense reconstruction point cloud for the three datasets. For ATE evaluation, the ground truth data

(a) *npuchangan*



(b) *npuyouyi*

(c) *famensi*

Fig. 6. Entire dense reconstruction results comparison. From left to right, we have the results of COLMAP, DenseFusion, and our method. (a) represents *npuchangan* dataset, (b) represents *npuyouyi* dataset, (c) represents *famensi* dataset.

Fig. 7. Comparing reconstruction results in Different situations. The left cannot acquire submap's relative transformation without the use of VGMN. The middle is a lack of center UAV image data with the use of VGMN. The right involves entire dataset with the use of VGMN.

TABLE IV

THE ATE OF DIFFERENT DATASETS. RMSE(M) VALUES ARE REPORTED.

Dataset	***npuyouyi***	***npuchangan***	***famensi***
ORB-SLAM2	1.016	1.220	1.068
ours (no VGMN)	0.459	0.561	0.487
ours (with VGMN)	**0.326**	**0.443**	**0.349**

TABLE V

THE ERROR OF DENSE RESTRUCTION COMPARED TO COLMAP. RMSE(M) VALUES ARE REPORTED.

Dataset	***npuyouyi***	***npuchangan***	***famensi***
RMSE (m)	2.486	4.466	2.091

is obtained via RTK recorded at the time of image capture. We also compare the results with running three instances of ORB-SLAM2 [39], as shown in Table IV. Compared to ORB-SLAM2, our system without VGMN achieves an ATE RMSE of 0.45 m in 3 km^2 aeras. Our system's dense reconstruction result with VGMN produces a 0.32 m RMSE of ATE, and the trajectory is as demonstrated by Figure 5(b). To assess the error in the reconstructed point cloud, we use the point cloud generated by COLMAP as the reference and employ CloudCompare software for point cloud registration and evaluation as presented in Table V. The experiments without using VGMN, with partial data, and with complete data are as shown in the Figure 7. Compared to COLMAP, our system provides dense construction result with a construction RMSE of less than 4.5 m. Within a reconstruction area of 1500 m $\times$ 2000 m, the average error of the dense reconstruction results is less than 0.31% of the actual scale.

V. CONCLUSIONS

In this paper, we propose a system called *AutoFusion* for large-scale multi-UAV dense reconstruction in real-time. The system focuses on robustness in large-scale scenarios, loss of geographical location, and weak co-visibility of images. We present a Visual Geolocation and Matching Network, which uses a graph convolutional neural network to acquire the image's geolocation information. In Multi-UAV SLAM system, the server includes the multi-agent weight BA optimization, the pose graph optimization with geolocation, and dense map fusion. Experimental results demonstrate that our system can complete dense reconstruction robustly and efficiently. In the future, we would like to support various front-end VO and VIO as agent for large-scale multi-UAV dense construction.

REFERENCES

[1] Z. Lai, F. Liu, S. Guo, X. Meng, S. Han, and W. Li, "Onboard real-time dense reconstruction in large terrain scene using embedded uav platform," *Remote Sensing*, vol. 13, no. 14, p. 2778, 2021.

[2] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, "Using unmanned aerial vehicles (uav) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments," *Remote sensing*, vol. 5, no. 12, pp. 6880–6898, 2013.

[3] P. Mandikal and V. B. Radhakrishnan, "Dense 3d point cloud reconstruction using a deep pyramid network," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1052–1060.

[4] M. Chen, Y. Tang, X. Zou, K. Huang, L. Li, and Y. He, "High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm," *Optics and Lasers in Engineering*, vol. 122, pp. 170–183, 2019.

[5] Y. Pan, Y. Dong, D. Wang, A. Chen, and Z. Ye, "Three-dimensional reconstruction of structural surface model of heritage bridges using uav-based photogrammetric point clouds," *Remote Sensing*, vol. 11, no. 10, p. 1204, 2019.

[6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[7] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[8] A. C. Jiménez, V. García-Díaz, R. González-Crespo, and S. Bolaños, "Decentralized online simultaneous localization and mapping for multi-agent systems," *Sensors*, vol. 18, no. 8, p. 2612, 2018.

[9] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[10] V. Kachurka, B. Rault, F. I. I. Muñoz, D. Roussel, F. Bonardi, J.-Y. Didier, H. Hadj-Abdelkader, S. Bouchafa, P. Alliez, and M. Robin, "Weco-slam: Wearable cooperative slam system for real-time indoor localization under challenging conditions," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5122–5132, 2021.

[11] Y. Yue, C. Yang, Y. Wang, P. C. N. Senarathne, J. Zhang, M. Wen, and D. Wang, "A multilevel fusion system for multirobot 3-d mapping using heterogeneous sensors," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1341–1352, 2019.

[12] H. Shen, Q. Zong, B. Tian, and H. Lu, "Voxel-based localization and mapping for multirobot system in gps-denied environments," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 10, pp. 10 333–10 342, 2022.

[13] N. Mahdoui, V. Frémont, and E. Natalizio, "Communicating multi-uav system for cooperative slam-based exploration," *Journal of Intelligent & Robotic Systems*, vol. 98, pp. 325–343, 2020.

[14] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM–collaborative visual-inertial slam," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2762–2769, 2018.

[15] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "COVINS: Visual-inertial slam for centralized collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2021, pp. 171–176.

[16] M. Patel, M. Karrer, P. Bänninger, and M. Chli, "COVINS-G: A generic back-end for collaborative visual-inertial slam," *arXiv preprint arXiv:2301.07147*, 2023.

[17] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[18] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[19] X. Wang, C. Olston, A. D. Sarma, and R. Burns, "CoScan: cooperative scan sharing in the cloud," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, 2011, pp. 1–12.

[20] Y. Chang, Y. Tian, J. P. How, and L. Carlone, "Kimera-Multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 210–11 218.

[21] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[22] X. Liu, W. Ye, C. Tian, Z. Cui, H. Bao, and G. Zhang, "Coxgraph: multi-robot collaborative, globally consistent, online dense reconstruction system," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8722–8728.

[23] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, "Deep local shapes: Learning local sdf priors for detailed 3d reconstruction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 2020, pp. 608–625.

[24] T. Sattler, B. Leibe, and L. Kobbelt, "Exploiting spatial and co-visibility relations for image-based localization," *Large-Scale Visual Geo-Localization*, pp. 165–187, 2016.

[25] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[26] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation techniques: principles and applications*. Springer Science & Business Media, 2012.

[27] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2011.

[28] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[29] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[30] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 141–14 152.

[31] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[32] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," 2008.

[33] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[34] J. G. Manchuk and C. Deutsch, "Conversion of latitude and longitude to utm coordinates," *Paper 410, CCG Annual Report*, vol. 11, 2009.

[35] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[36] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv preprint arXiv:1901.03638*, 2019.

[37] L. Chen, Y. Zhao, S. Xu, S. Bu, P. Han, and G. Wan, "Densefusion: Large-scale online dense pointcloud and dsm mapping for uavs," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4766–4773.

[38] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian conference on computer vision*. Springer, 2010, pp. 25–38.

[39] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.