Semantic 3D Reconstruction from Multi-Robot Systems

Shane Dirksen*
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, California
shanedirksen@ucsb.edu

Ron Kibel*
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, California
rkibel@ucsb.edu

***Abstract*—For many multi-robot applications, reliable perception and mapping are crucial for exploration, navigation, and search and rescue. Conventional multi-view stereo (MVS) and structure from motion (SfM) pipelines can be effective for 3D reconstruction, but they often struggle under challenging conditions (motion blur, sensor noise, weather) and lack a semantic understanding of their environment. In this paper, we propose an end-to-end multi-robot perception framework that fuses both geometric and semantic cues by modeling the robots as a graph, with edges denoting modes of communication. Each robot processes its monocular image with a shared feature extractor, then exchanges latent features (including spatial encoding) with neighboring robots via a message-passing Graph Attention Network (GAT). As a result, we produce an updated representation for each drone that integrates local features with relevant cues from other viewpoints, providing a more robust prediction for our specific tasks: depth prediction and semantic segmentation. Using a masked-attention approach based on drone-to-drone distance, we observe an improvement in training time compared to traditional cross-attention.**

***Index Terms*—multi-robot, 3D reconstruction, semantic segmentation, graph attention network (GAT), cross-attention**

I. Introduction

Multi-robot 3D reconstruction and scene understanding has emerged as a critical capability in advanced field robotics over the last decade. Specifically in areas like autonomous navigation, search and rescue, and environment monitoring missions, multi-robot schemes like UAV swarms or ground robot teams have proved vital in quickly covering large areas, while also providing redundancy through multiple sources of feedback. In such scenarios, reliable perception and mapping are highly important—robots must learn to build accurate models of unknown terrain, navigate safe paths, or identify specific targets.

To carry out such tasks, classical techniques such as multi-view stereo and structure-from-motion have long provided geometry-based reconstruction pipelines. These methods can recover geometry from a set of images by using low-level visual cues (feature tracking) to infer the perspective of each image. However, while effective under ideal conditions, MVS and SfM are often limited by their overreliance on purely geometric cues and lack the semantic context or robustness required when dealing with challenging environmental conditions (weather, lighting) or with sensor noise.

* Both authors contributed equally to this research.

Recent advancements in deep learning (DL), specifically in computer vision (CV) have trickled into the robotics sector, and now routinely improve on such classical methods in visual identification tasks like depth estimation and semantic segmentation. These new algorithms motivate merging high and low-level feature understanding, which results in a more robust and semantically-aware visual processing model. However, they encounter substantial scalability challenges when adapted to multi-robot scenarios. Traditional DL architectures frequently use a fully-connected cross-attention mechanism between robots for centralized feature fusion, but this scales poorly.

In this work, we propose a decentralized multi-robot 3D reconstruction framework that leverages a Graph Attention Network, a subset of Graph Neural Network (GNN). Modeling the multi-robot as a graph problem, where robots (nodes) can send messages to each other across communication links (edges), we consider how performance changes when applying a geometry-implicit DL model as a communication mechanism. Specifically, we aim to analyze how limiting the connectedness of our network based on cross-attention (fully-connected robot-to-robot interactions) and thresholded masked attention (interaction limited by robot-to-robot distance) will affect performance metrics on depth estimation and semantic segmentation tasks, as well as processing time.

The paper is organized as follows. Section II is a brief literature review of similar research. Section III describes the architecture for which we pose the multi-robot problem and our training and evaluation protocols. Section IV presents the methods for collecting our train, evaluation, and test datasets. Section V discusses quantitative and qualitative results from our experiments. Section VI presents relevant findings from our work. Lastly, Section VII discusses expansions to this project for a potential future publication.

II. Related Works

Classical multi-robot 3D reconstruction schemes have historically relied on multi-view stereo and structure-from-motion techniques, where low-level visual cues are mapped from one image to the next, providing a sense of perspective [2] [3]. Research has highlighted their effectiveness in controlled environments and their ability to capture essential details, but these methods are constrained by their sole reliance on these low-level details, making them vulnerable to adverse

conditions such as sensor noise, motion blur, weather, or extreme illumination.

Advances in deep learning have substantially improved perception capabilities, particularly in semantic segmentation and depth estimation tasks. Long et al. [10] first proposed fully convolutional networks for semantic segmentation, which were further enhanced by architectures such as U-Nets [12] and DeepLab [1]. Monocular depth estimation also benefited significantly from CNN-based methods: Godard et al. [4] demonstrates improvements over classical methods by incorporating learned contextual cues.

In multi-robot scenarios, the performance of these DL approaches can be further improved by enabling robots to exchange perceptual information after feature extraction. Who2com [9] proposes a multi-stage communication mechanism that allows robots to request and send information over a limited bandwidth network; their later work, When2com [8] further evaluates this mechanism by clustering robots across communication groups. The proposed task is semantic segmentation, but similar goals exist for robust depth estimation [2][3].

While these approaches show the feasability of communication in limited bandwidth scenarios, they all adopt fully-connected cross-attention mechanisms for feature fusion, which hinders scalability as swarm sizes increase. To address this challenge, Graph Neural Networks, particularly Graph Attention Networks [13] have recently gained attention, as they incorporate learned attention scores—and therefore filtering—in the message-passing mechanism. Zhou et al. [14] confirms the effectiveness of GATs in multi-robot communication mechanisms, noting substantial improvements in segmentation accuracy and depth estimation across a set of five drones. However, there is limited analysis of the difference in computation between cross-attention and masked (filtered) attention, specifically in a larger swarm.

III. Methods

We propose a collaborative aerial perception framework using Graph Attention Networks for information sharing among multiple drones operating in a swarm formation. This framework addresses monocular depth estimation and semantic segmentation, allowing drones to maintain perception when sensors are corrupted or malfunctioning.

We deploy a 16 drone dataset, each with an RGB camera. Each drone processes its local observations using a shared neural network, then exchanges information with nearby drones via graph-based communication. This approach integrates multiple viewpoints, which is useful when some drones have sensor degradation.

A. Architecture

Our architecture has four components:

1) A feature extraction backbone,
2) A graph formation module,
3) A GAT for information sharing,
4) Task-specific decoders for depth estimation and semantic segmentation.

1) Feature Extraction: Each drone uses a MobileNetV2 backbone (pretrained on ImageNet) to extract features from 224×224 RGB images. This backbone is lightweight yet retains sufficient representational capacity for our tasks.

2) Graph Formation: Each drone is represented as a node in a spatial graph. We create an edge between two drones if they lie within a variable distance threshold. We test across several thresholds, including distances of 12, 25, and 50 units. With a 12-unit threshold, the graph is sparsely connected, as most nodes only have a few direct connections. At 25 units, the graph becomes locally dense, where each node is well-connected to nearby neighbors but does not form a fully connected structure (see Fig. 1). At 50 units, the graph is highly connected, with most nodes having edges to nearly all others. Each edge also has a relative pose encoding that captures their positional and orientational relationship.

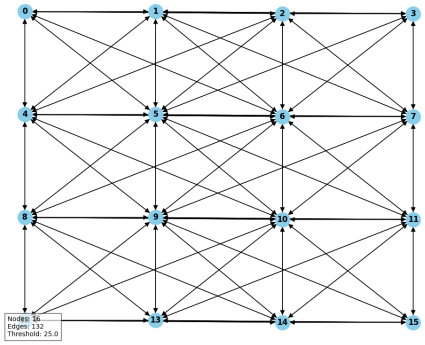

Fig. 1. Graph connectivity at 25-unit threshold, showing local density.

We employ a GAT with 8 attention heads. It processes node features (drone observations) alongside edge features (relative poses). The attention coefficients are computed as

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})},$$

where e_{ij} indicates how important drone j's features are to drone i. Our GAT has a hidden dimension of 128 and an output dimension of 64, with a dropout rate of 0.1. After the attention operation, each drone's feature representation is updated with information from its neighbors.

3) Task-Specific Decoders:

a) Depth Decoder.: A set of upsampling layers converts the GAT output back to 224×224 resolution, yielding a single-channel depth map.

b) Segmentation Decoder.: A similar upsampling design uses transposed convolutions to produce multi-channel outputs for semantic classes (buildings, road, vegetation, vehicles, background).

B. Training

We employ a multi-task loss to jointly optimize depth estimation and semantic segmentation:

$$\mathcal{L} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{seg}},$$

where

$$\mathcal{L}_{\text{depth}} = \frac{1}{N}\sum_{i=1}^{N} \left| y^i_{\text{depth}} - \hat{y}^i_{\text{depth}} \right|, \quad \mathcal{L}_{\text{seg}} = \text{CrossEntropyLoss}.$$

We train for 30 epochs using the Adam optimizer (learning rate = 0.001, weight decay = 10^{-5}). We experimented with multiple epochs and found the improvement between 30-100 epochs was minimal. The batch size is 4. We split the dataset into training (75%), validation (15%), and test (10%).

C. Corruption-Robust Training

To improve robustness, we randomly corrupt 0% to 50% of the drone inputs during training with motion blur or shot noise (intensity levels 1, 3, and 5). This encourages the network to rely on uncorrupted drones and handle degraded inputs.

D. Evaluation

We evaluate the system under three corruption scenarios: none, partial (33% corrupted), and full (all corrupted).

For depth estimation, we use absolute relative error, which measures the average relative difference between predicted and ground truth depths, squared relative error, which places more weight on larger errors by squaring the differences, and root mean squared error (RMSE), which captures overall prediction accuracy by penalizing larger deviations.

For semantic segmentation, we measure mean Intersection over Union (mIoU), which quantifies how well predicted segmentations overlap with ground truth labels across all classes, and pixel accuracy, which reflects the percentage of correctly classified pixels in the entire image.

We also compare against a cross-attention baseline that uses direct attention across all drones. This highlights the efficiency and scalability of the graph-based approach, especially as the number of drones increases.

IV. Dataset Collection

For extracting multi-drone data, we first toyed with NVIDIA Omniverse Isaac Sim, a robotics simulator and synthetic data generator. However, trying to simulate multiple camera trajectories (a "drone swarm") and extract ground truth depth and segmentation proved difficult. Moreover, the environment (included in figure 2) was not as complicated as we would have liked and did not fit the use cases we described previously.

Instead, we transitioned to the Microsoft AirSim simulator, built on Unreal Engine and tailored for aerial and ground vehicles. AirSim proved especially well-suited for our tasks, as we could simulate a large number of drones in flight at once, extract ground-truth depth maps as well as segmentation maps for pure RGB video feed, and customize the environment according to a number of visibility parameters. We chose to launch our drone trajectories in the Neighborhood (NH)

Fig. 2. Isaac Sim warehouse environment. A) Aerial bird's-eye view with camera pose. B) Rendered environment.

pre-built environment (see Figure 3), as it proved sufficiently complex to test intricate geometric patterns and resembled a real-world dataset.

Fig. 3. Microsoft AirSim Neighborhood environment, aerial bird's-eye view.

We simulated 16 drones in total, flying in a 4-by-4 grid formation around a central trajectory that follows a 1:2 Lissajous curve (a parametric waveform resembling a figure-8) plotted in figure 4. There is slight height variation to this central trajectory. As the drones fly, the grid itself rotates, and the yaw, roll, and pitch of each drone change variably so as to expose each drone to different angles of the environment

and prevent overreliance on a specific perspective.

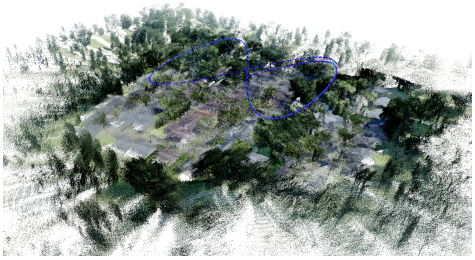

Fig. 4. Drone center-of-mass trajectory. The plotted path follows a 1:2 Lissajous curve, akin to a figure-8, with slight z-variation.

As a result of this simulation, a set of several thousand images is collected, consisting of pure RGB, ground truth depth, and ground truth segmentation for each drone and each time frame. We then use the robustness library retrieved from Hendrycks and Dietterich [6] to artificially corrupt the pure RGB images of each drone according to two types of noise: motion blur (mb), and motion blur combined with shot noise, a type of electronic noise (mb+sn). Moreover, we generate these corruptions at 3 different intensities—i1, i3, and i5—as shown in figure 5. This data collection method produces 22K RGB images from different perspectives with 6 different types of noise, as well as ground truth depth, segmentation, and precise environment coordinates.

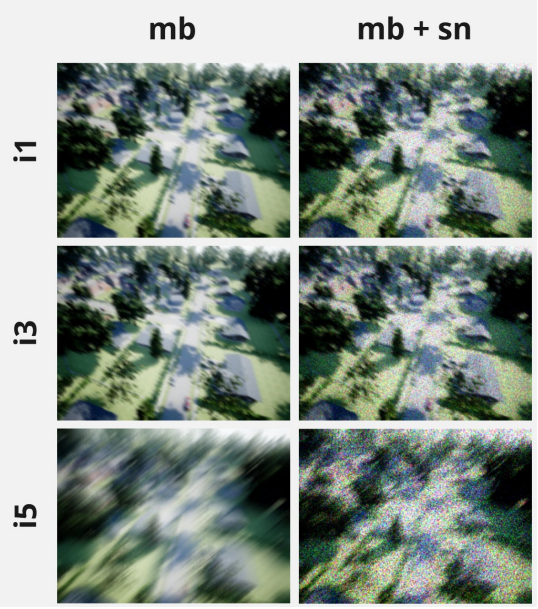

Fig. 5. Example corruptions motion blur (mb) and motion blur combined with shot noise (mb+sn) at various intensities, derived from the robustness library [6].

V. Results

We evaluated our proposed multi-drone perception framework using both quantitative metrics and qualitative visualizations. Experiments were conducted using a fleet of 16 drones, comparing our GAT approach against a Cross-Attention baseline. Both models were trained for 30 epochs across various distance thresholds (12, 25, and 50 units) to investigate the impact of connectivity density on performance and computational efficiency.

Table I presents a comprehensive comparison between the GAT and Cross Attention models across different distance thresholds.

TABLE I
PERFORMANCE COMPARISON FOR GAT AND CA (CROSS ATTENTION) ACROSS DISTANCE THRESHOLDS. LOWER IS BETTER FOR ABS REL, SQ REL, RMSE; HIGHER IS BETTER FOR MIOU, PIXEL ACC.

Model	Thresh.	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑	Pixel Acc↑	Time/Epoch (s) [Total (s)]
CA	12	0.903	6218.65	12522.15	0.696	0.823	51.03 [1530.85]
GAT	12	1.052	10738.64	11862.31	0.694	0.821	**47.60 [1428.12]**
CA	25	0.521	4855.62	12174.90	0.699	0.824	69.99 [2099.63]
GAT	25	0.572	5783.03	12465.61	0.700	0.825	**56.96 [1708.80]**
CA	50	0.658	6064.14	12213.12	0.704	0.826	82.34 [2470.33]
GAT	50	0.938	6314.64	12527.47	0.684	0.814	**72.16 [2164.77]**

Our qualitative results (Figure 6) further support these findings, showing that both models produce visually similar outputs despite their architectural differences, with the GAT model maintaining high-quality predictions while requiring less computational resources. Both models remain robust even under high levels of input corruption. The predicted segmentation and depth maps, while abstract, successfully capture general structures such as trees, buildings, and roads.

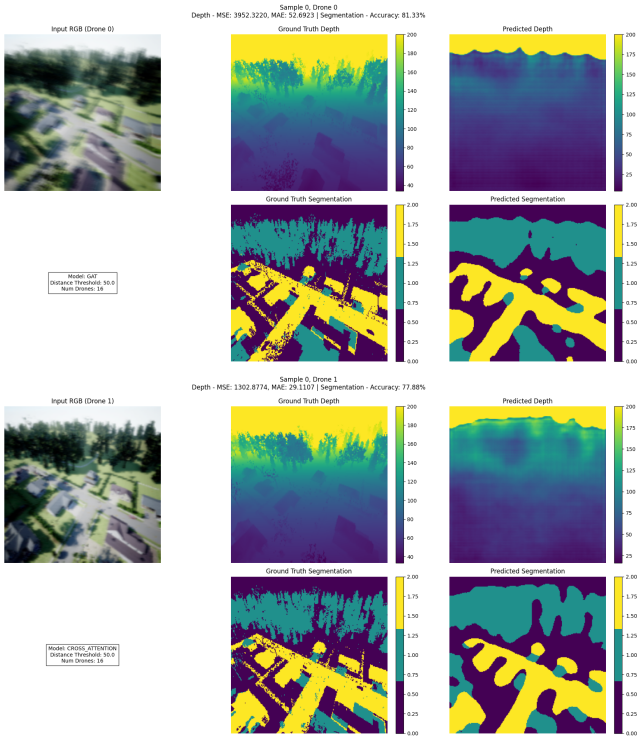

Fig. 6. Visual comparison of depth and segmentation predictions from GAT (top) and Cross Attention (bottom) models at distance threshold 50. The figure shows RGB input (left), ground truth (middle), and model predictions (right).

VI. Conclusion

Our results demonstrate that GAT achieves comparable performance to Cross Attention while being computationally more efficient. Across different distance thresholds, GAT consistently produced high-quality depth and segmentation predictions with reduced processing time.

GAT does not outperform Cross Attention because long-range dependencies provide little benefit in this scenario. In multi-drone perception, the most relevant information comes from nearby drones with overlapping RGB data, while distant drones contribute little. As a result, GAT naturally focuses on local neighbors, just as Cross Attention does, leading to similar performance.

The computational advantage of GAT comes from its sparser attention mechanism and fewer matrix operations. Unlike Cross Attention, which requires every query to attend to all keys in the input, GAT restricts attention to local neighborhoods. This eliminates the quadratic scaling of attention computation, making it more efficient for multi-robot systems where long-range dependencies are not critical.

Additionally, GAT directly computes attention scores using learned weights for each node and its neighbors, avoiding the full pairwise attention calculations required in Cross Attention. This reduces memory usage and speeds up inference, making GAT a more scalable choice for multi-drone perception tasks.

Our findings suggest that for distributed multi-robot perception, GAT provides a more efficient communication mechanism without sacrificing accuracy.

VII. Future Work

We believe that this project provides a strong foundation for experimenting with GNN architectures in simulated multi-robot environments. Following this work, there are a vast number of directions to further the contributions of this project. For one, our chosen simulator environment, AirSim, provides inbuilt weather and illumination customization (fog, frost, snow, dust), and it would be interesting to explore how cross-attention compares with masked attention for these types of environment changes. Namely, if we could identify specific weather patterns that have considerable difference (likely declines in accuracy) it could serve as a relevant exploration. Furthermore, altering the Graph Attention Network beyond distance thresholding (e.g. varying the number of attention heads) could also provide further relevance to our applications.

Some other future topics for our work include extended scalability (how GAT performs in an environment with significantly more than 16 drones), generalizability (whether this model can generalize to other AirSim environments with different structures) and Sim-to-Real (whether this model can work for real-world datasets).

Above all, we believe the most significant contribution to turn this project into a publishable paper would be to consider other GNN approaches. Although GAT reduced computation time, it compared similarly to cross-attention depth and segmentation accuracy metrics. GAT is inherently used to model long-distance dependencies, but in this case we only consider a robot's closest neighbors; as such, it would make sense to also use other GNNs like Graph Convolutional Networks [7] and GraphSAGE [5]. To focus more on computation, proposing a solution with highly efficient Graph Transformers like Graph GPS [11] could also be a point of consideration.

VIII. Team Contributions

Ron Kibel primarily focused on the dataset generation while **Shane Dirksen** primarily focused on model training. Both were involved in model development and research. The model and training repository available at https://github.com/shanedirksen/droneGAT, and the dataset generation repository is available at https://github.com/rkibel/AirSim.

References

[1] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *CoRR* abs/1606.00915 (2016). arXiv: 1606.00915. URL: http://arxiv.org/abs/1606.00915.

[2] A. Dias et al. "Multi-robot cooperative stereo for outdoor scenarios". In: *2013 13th International Conference on Autonomous Robot Systems*. 2013, pp. 1–6. DOI: 10.1109/Robotica.2013.6623531.

[3] Anton Filatov, Mark Zaslavskiy, and Kirill Krinkin. "Multi-Drone 3D Building Reconstruction Method". In: *Mathematics* 9.23 (2021). ISSN: 2227-7390. DOI: 10.3390/math9233033. URL: https://www.mdpi.com/2227-7390/9/23/3033.

[4] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. "Unsupervised monocular depth estimation with left-right consistency". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 270–279.

[5] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *CoRR* abs/1706.02216 (2017). arXiv: 1706.02216. URL: http://arxiv.org/abs/1706.02216.

[6] Dan Hendrycks and Thomas G. Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *CoRR* abs/1903.12261 (2019). arXiv: 1903.12261. URL: http://arxiv.org/abs/1903.12261.

[7] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *CoRR* abs/1609.02907 (2016). arXiv: 1609.02907. URL: http://arxiv.org/abs/1609.02907.

[8] Yen-Cheng Liu et al. "When2com: Multi-Agent Perception via Communication Graph Grouping". In: *CoRR* abs/2006.00176 (2020). arXiv: 2006.00176. URL: https://arxiv.org/abs/2006.00176.

[9] Yuchen Liu et al. "Who2Com: Collaborative Perception via Learnable Handshake Communication". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 6876–6883.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[11] Ladislav Rampášek et al. *Recipe for a General, Powerful, Scalable Graph Transformer*. 2023. arXiv: 2205.12454 [cs.LG]. URL: https://arxiv.org/abs/2205.12454.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[13] Petar Velickovic et al. “Graph Attention Networks”. In: *International Conference on Learning Representations (ICLR)* (2018).

[14] Yang Zhou et al. “Multi-Robot Collaborative Perception with Graph Neural Networks”. In: *CoRR* abs/2201.01760 (2022). arXiv: 2201.01760. URL: https://arxiv.org/abs/2201.01760.