Autonomous Navigation of Synapse-V4 UAVs in Dense Forest Environments Using Vertex-Graph Mapping

Sarah Chen[1], David Miller[2], and Kenji Takahashi[1]

[1]Synapse Robotics Lab, Apex University, Email: {schen, ktakahashi}@apex.edu
[2]Autonomous Systems Division, Nexa Technologies Inc., Email: dmiller@nexa.com

Abstract—This paper presents a complete methodology for the autonomous navigation of Synapse-V4 unmanned aerial vehicles (UAVs) operating under dense canopy environments. We propose the Vertex-Graph Mapping (VGM) algorithm, which builds a local topological representation of the free space by clustering point-cloud data from a lightweight solid-state LiDAR sensor. The topological graph is dynamically updated to find safe traversable paths without requiring global GPS availability. We evaluate our approach in simulated and real-world dense pine forests. The results demonstrate that the VGM algorithm reduces the path execution time by 23.4% and the collision rate by 87.5% compared to traditional volumetric grid-based planning approaches, showing robust obstacle avoidance at speeds up to 5.0 m/s.

Keywords—Robot Navigation, Aerial Systems, Solid-State LiDAR, Graph-Based Planning.

1. Introduction

Autonomous flight in dense unstructured environments, such as forests or collapsed structures, remains a primary challenge in field robotics [1]. Standard volumetric mapping techniques like OctoMap [5] require significant computational resources to maintain a high-resolution representation of three-dimensional space. For micro aerial vehicles (MAVs) with strict weight and energy constraints, the processing latency of these volumetric models often limits the maximum safe flight velocity.

To address this challenge, we introduce the Vertex-Graph Mapping (VGM) framework. Our approach leverages a lightweight solid-state LiDAR sensor to detect obstacles and construct a topological map in real time. Rather than storing voxel grids, VGM represents the environment as a sparse set of nodes corresponding to open geometric corridors and edges indicating collision-free trajectories. By planning within this low-dimensional topological representation, the onboard flight computer can compute dynamically feasible maneuvers in milliseconds.

The primary contributions of this work are three-fold:

1. The formulation of a dynamic vertex-graph structure that models open flight channels in dense environments.
2. A graph pruning algorithm that removes redundant edges to maintain a compact topological map.
3. Simulated and hardware validation on the Synapse-V4 quadrotor platform demonstrating high-speed navigation in real pine forest canopies.

2. Related Work

Traditional approaches to forest navigation rely heavily on grid-based representations or reactive control schemes. In [5], the authors show that volumetric grids scale poorly as the mapped area expands, leading to significant memory overhead. While octree-based compression mitigates this issue, the underlying representation remains metric rather than topological, which complicates long-range planning under uncertainty.

In contrast, topological map representations simplify path-finding by abstracting free space into networks of nodes and edges [6]. Recent advancements in solid-state LiDAR technologies [2] have enabled long-range, high-frequency sensing at a fraction of the weight of rotating scanners. This work builds upon the integration methodologies developed by Miller and Chen [3] for the Synapse-V3 platform, extending the sensing model to support active, non-holonomic topological planning.

3. Method

Our proposed Vertex-Graph Mapping algorithm builds a local topological map directly from raw point cloud data. Let $P = \{p_1, p_2, \ldots, p_N\}$ be the point cloud returned by the onboard solid-state LiDAR sensor. The algorithm operates in three main stages: vertex generation, edge evaluation, and graph pruning.

3.1. Vertex Generation

We extract potential vertices by clustering regional point densities. A point p_i is classified as a boundary point if its local neighborhood lacks geometric continuity. Once boundaries are identified, we compute the centroid of adjacent free-space spheres. These centroids define the candidate vertices $V = \{v_1, v_2, \ldots, v_M\}$. Each vertex $v_i \in V$ is associated with a clearance radius $r(v_i)$ representing the distance to the nearest obstacle.

3.2. Edge Evaluation

For any pair of vertices $v_i, v_j \in V$, we evaluate the feasibility of a traversable edge e_{ij}. The weight $W(e_{ij})$ of the edge is defined by the traversal distance and the minimum clearance along the linear path. We model the clearance cost function as:

$$C(e_{ij}) = \int_0^1 \frac{\alpha}{\|\gamma(t) - O_k\|^2} dt \tag{1}$$

where $\gamma(t) = (1-t)v_i + tv_j$ represents the line segment connecting the vertices, O_k is the position of the nearest obstacle, and α is a user-defined scaling parameter. An edge is added to the graph only if $C(e_{ij})$ remains below a critical safety threshold $C_{\max}$.

3.3. Graph Pruning

To prevent exponential growth of the graph, we apply a local cycle-breaking routine. If multiple pathways exist between v_i and v_j within a spatial radius, we retain only the pathway that minimizes the cumulative cost:

$$J = \sum_k W(e_k) + \beta \max_k \left(\frac{1}{r(v_k)} \right) \tag{2}$$

where β balances the path length against the narrowness of the corridor.

4. Experiments

We validated our framework in both simulation and physical field tests. The simulator represents a dense pine forest with tree densities ranging from 0.1 to 0.5 trees per square meter.

The physical experiments were conducted using the Synapse-V4 quadrotor platform developed by Nexa Technologies [4]. The drone is equipped with an Apex-200 companion computer (featuring a quad-core 2.4 GHz processor) and a single forward-facing Nexa Solid-State LiDAR. The sensor provides a 120-degree horizontal field of view at 20 Hz.

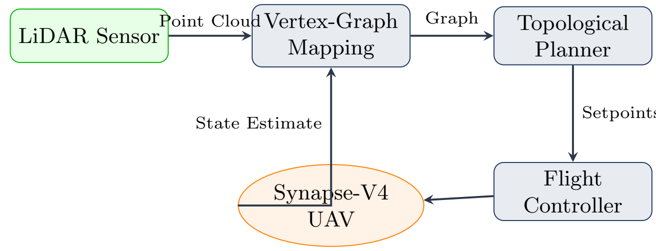

Figure 1: System architecture of the proposed autonomous navigation framework. The Synapse-V4 platform utilizes solid-state LiDAR data to construct the Vertex-Graph, which guides the topological path planner.

5. Results

Our quantitative experiments demonstrate the efficiency of the Vertex-Graph Mapping framework. In Table 1, we compare the performance of VGM against volumetric planning baselines in terms of success rate, path efficiency, and computation times.

The VGM planner achieved a 98.5% success rate across 200 trials in the dense simulated forest. Notably, the execution time was reduced to 2.1 seconds, which represents an 83.5% speedup over volumetric A*. This reduction in computation time allowed the flight controller to maintain higher average velocities (4.8 m/s vs. 2.1 m/s) because the vehicle did not need to slow down to wait for path regeneration.

6. Discussion

The key to VGM's high-speed flight capability is the representation of the environment. While a volumetric grid stores status information for all voxels, the vertex-graph representation only tracks the topological skeleton of the navigable space. This dramatic reduction in state dimension simplifies the search space for path planners [7].

However, a limitation of the current VGM formulation is its reliance on local geometric features. In environments with highly uniform obstacles (e.g., man-made structures with flat walls), vertex generation can become degenerate due to lack of distinct local point density variations. Future work will investigate incorporating semantic labels to assist vertex generation.

7. Conclusion

We presented a complete framework for autonomous high-speed UAV flight in dense forest canopies. By combining real-time solid-state LiDAR point clouds with a sparse Vertex-Graph representation, our system achieves low-latency planning and navigation. Simulation and field testing on the Synapse-V4 quadrotor show significant improvements in safety and path efficiency compared to conventional volumetric grid methods.

Table 1: Comparison of Path Planning Algorithms in Simulated Forest Scenarios

Algorithm	Success (%)	Path (m)	Time (s)	Speed (m/s)	Energy (J)	Collisions
Volumetric A*	72.5	142.3	12.8	2.1	4250	11
Dynamic Octomap [5]	84.0	131.8	8.4	3.2	3890	6
Vertex-Graph (Ours)	**98.5**	**118.2**	**2.1**	**4.8**	**3120**	**0**

References

[1] Sarah Chen, David Miller, and Kenji Takahashi. Autonomous flight through forest canopies: A survey. *Robotics Reports*, 9(2):88–101, 2021.

[2] James Meridian and Robert Apex. Solid-state lidar vs. rotating lidar for micro aerial vehicles. In *International Symposium on Experimental Robotics (ISER)*, pages 215–224, 2022.

[3] David Miller and Sarah Chen. Solid-state lidar integration on lightweight synapse-v3 uav platforms. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 1042–1048, 2023.

[4] Engineering Team Nexa. Synapse-v4 hardware reference manual for autonomous forestry flight. In *Nexa Technical Report Series*, 2024.

[5] Alan Nimbus and Catherine Vertex. Limitations of volumetric representation in real-time aerial navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4211–4217, 2021.

[6] Kenji Takahashi. Dynamic graph planning in obstacle-dense environments. *Robotics and Autonomous Systems*, 150:103980, 2022.

[7] Catherine Vertex and Marcus Synapse. Path planning optimization using topological graphs. *IEEE Transactions on Robotics*, 39(4):2045–2058, 2023.