Multi-Agent Autonomous Mapping of Unknown GPS-Denied Environments Using a Relative Navigation Framework

Jacob M. Olson[1], Timothy W. McLain[2] [TODO: include Matthiew Labbe?]

***Abstract*— [TODO: When generating 3D maps with unmanned aerial vehicles (UAVs), it is important for the mapping algorithm to have good coverage of the environment. It is also important, especially when planning paths for multiple agents, to have loop closures along each flight path and with other agents. Because multirotor UAVs are limited in flight time, the flight paths must be limited in length. Generating a good flight path to map a new environment can be difficult and tedious because of the free-form nature of a flight path. To solve this problem, we propose using a genetic algorithm designed to maximize total area coverage while minimizing flight time and enforcing sufficient loop closures. The natural ability of genetic algorithms to rapidly explore a design space is advantageous when solving complex free-form problems like path planning.]**

I. Introduction

Autonomously generating an internal 3D map of a building requires intelligent control of the mapping vehicle. This ranges from manually planning a flight path with simple obstacle avoidance to full-stack exploration and path planning algorithms.

Generating a flight path is a nontrivial task, especially when multiple objectives are considered. For example, the algorithm may want to maximize observed coverage of the space while minimizing flight time. This becomes tedious with each new space and more difficult as the complexity of the space increases with more rooms and inter-connecting hallways. When planning paths to achieve good coverage of the area rather than obstacle avoidance or traversability, it becomes much more complex. To plan effective flight paths in more complex flight spaces, using algorithms becomes a necessity.

When generating paths for the purpose of building a 3D map of the environment in a GPS-denied area such as indoors, enforcing loop closures becomes critical. Loop closures both between paths of different agents and between different segments of the same path are critical to successfully generate a usable map. Loop closures between different segments of the same path significantly reduce drift that occurs in visual odometry algorithms necessary for navigation and estimation with UAVs. Loop closures between paths of different agents ensures that the individual maps of each agent will combine into a single map.

There are several different approaches to coverage path planning that have been explored over recent years. These approaches range from 2D coverage where the robot must pass over all points in its known environment, common for uses such as cleaning robots or mine searching [1] to using coverage planning to generate 3D maps of outdoor terrain using viewpoint information from the robot's camera [2]. These approaches differ from the coverage planning desired for generating 3D maps of indoor environments with multiple UAVs in that they do not enforce loop closures in the generated paths.

The complexity of the design space renders many simple optimization routines improper for this problem. Further, by the structure of the problem, derivatives are not available for each design variable. Thus, a derivative-free approach is required. We chose to use a genetic algorithm because they are well suited to this type of problem. As outlined by Gen et al. [3], genetic algorithms are a powerful tool for solving complex, multi-objective optimization problems such as this. Genetic algorithms have been used in coverage path planning before, Yakoubi et al. details an approach to 2D coverage planning of a cleaning robot using a multi-objective genetic algorithm [4] and Hameed discussed using them for 3D terrain [5]. Similarly to [1], and [2], these approaches to not enforce loop closures and the objective is to visit all areas on the ground. Since the goal of this optimization is to get a path that when flown will generate a high fidelity 3D map of an indoor area, viewing all of the walls and obstacles is more important than visiting all areas on the ground. The genetic algorithm proposed in this paper takes these differences into consideration.

We first discretized and simplified the model as much as possible to shrink the design space before applying a genetic algorithm. We found our genetic algorithm was able to effectively plan single agent and multi-agent paths in an arbitrary environment and satisfy the constraints of the problem.

The remainder of the paper is organized as follows: Section **??** describes the method for simplifying the design space to create a reasonable problem for the genetic algorithm to solve. Section II describes the approach and architecture used to generate optimal paths for a single UAV, then the method is extended to multi-agent path planning. Results showing and evaluating the generated paths are presented in Section V. Finally, conclusions are presented in Section VI.

II. Technical Approach

A. Problem Statement

The goal of

B. Network Diagram

1) Map Generation:

2) Waypoints:

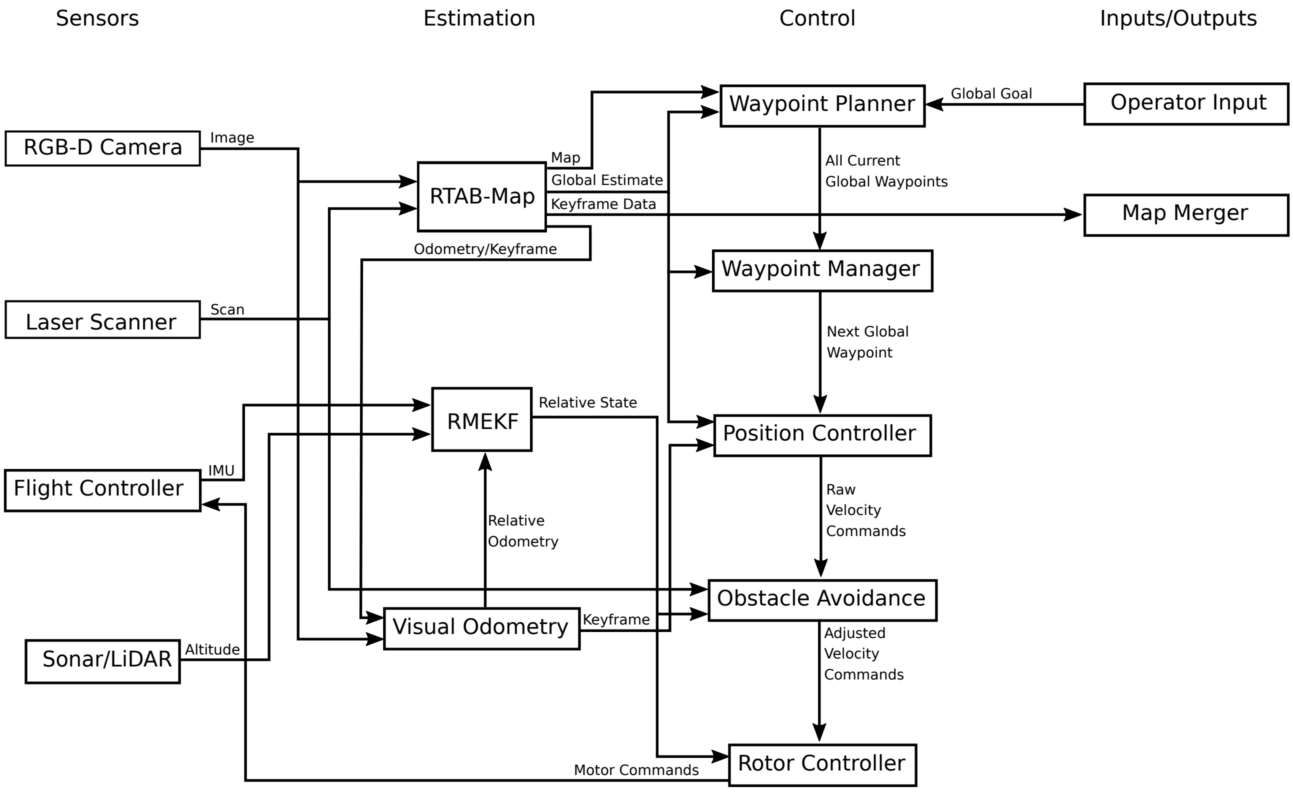

Fig. 1. The network diagram of the relative navigation framework proposed in this paper

C. *Objective Functions*

1) *Maximize Coverage:*

$$d_{view,i} = \max(\min(d_{max}, \min(d_{obs,i})), d_{min}) \tag{1}$$
$$i = 0...m$$

$$\mathcal{C} = \alpha_{cov}\mathcal{C}_{open} + \left(1 - \alpha_{cov}\right)\mathcal{C}_{walls}, \tag{2}$$

III. Planning

A. *Global Goal Following with Relative Estimation*

B. *Reactive Path Planning*

IV. Map Merging

V. Results and Discussion

The designer is required to set many parameters for the genetic algorithm to work as derived in [6]. The parameters we used, along with parameters specific to our implementation, are outlined in Table **??**.

VI. Conclusions

We were able to successfully generate paths with sufficient coverage to generate a high-fidelity map of the environment that do not appear to meander too much for a single UAV. Future work on this project will include improving the loop closure implementation to allow the algorithm to explore the design space better without getting caught in local minima. This will enable more efficient paths to be created. We expect the need to use the paths generated with a lower level path planner that can dynamically change waypoints according to obstacles and other unmapped disparities. Finally we plan to test the paths generated both in high fidelity simulation as well as in hardware.

References

[1] H. Choset, “Coverage for robotics-A survey of recent results,” Tech. Rep., 2001. [Online]. Available: https://link.springer.com/content/pdf/10.1023%2FA%3A1016639210559.pdf

[2] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, and R. Siegwart rsiegwart, “Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots,” vol. 40, pp. 1059–1078, 2016. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2Fs10514-015-9517-1.pdf

[3] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*. Wiley, 2000. [Online]. Available: https://www.wiley.com/en-us/Genetic+ Algorithms+ and+ Engineering+ Optimization-p-9780471315315

[4] M. A. Yakoubi and M. T. Laskri, “The path planning of cleaner robot for coverage region using Genetic Algorithms,” *Journal of Innovation in Digital Ecosystems*, vol. 3, no. 1, pp. 37–43, jun 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352664516300050#br000025

[5] I. A. Hameed, “Intelligent Coverage Path Planning for Agricultural Robots and Autonomous Machines on Three-Dimensional Terrain,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 965–983, jun 2014. [Online]. Available: http://link.springer.com/10.1007/s10846-013-9834-6

[6] A. Parkinson, *Optimization-Based Design*. Provo, Utah: BYU Academic Publishing, 2019.