UAV Path Planning using Aerially Obtained Point Clouds

Alec Pugh and Luke Bower
Dr. Richard Chapman and Dr. Saad Biaz
Auburn University

More information about our project!

Introduction

With the growing use of unmanned aerial vehicles (**UAVs**) for commercial and military operations, path efficiency remains an utmost concern for battery and time preservation. This paper presents a method for three-dimensional (**3D**) path planning using point clouds obtained from the USGS 3DEP (**United States Geological Survey 3D Elevation Program**) dataset via Open Topography. The path itself is obtained using the A* algorithm, with additional modifications implemented to account for path smoothing, UAV size, and energy consumption. We also introduce a collision avoidance method using the precomputed data to account for unforeseen obstacles not rendered within the point cloud. The method presented is designed specifically for point clouds obtained via LiDAR (**Light Detection and Ranging**) scans from aircraft, where cavities may be present underneath the surface layer. Simulations show the validity of this method.

Technologies

Software/Equipment	Reasoning
OpenTopography	Used to access all point cloud datasets for testing purposes. OpenTopography API was used to access datasets remotely.
UTM	Python library used to convert WGS84 encoded coordinates to latitude and longitude to send to the UAV.
LASzip	Used initially to convert .las files to ASCII file formats.
Open3D	Python visualization library used due to its available functionality for point cloud operations.
ARDU Pilot	Used for communication from computer to UAV and for simulated UAV testing purposes.
Mission Planner/MAVProxy	Simulation and transmission software that can control UAVs for realistic simulation testing connecting to ARDU Pilot over IP addresses.
Selenium	Python library used with Chrome to remotely navigate OpenTopography.

Objectives

- Create a 3D UAV path planning approach that works on U.S. LiDAR data available through USGS 3DEP / OpenTopography
- Incorporate drone size, collision avoidance, path smoothing, and energy consumption into model
- Develop functionality to automate process including converting waypoints to latitude/longitude for flight with GPS-equipped UAV

Methodology

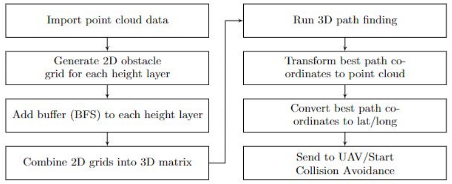

Figure 1: General overview of process.

Path Planning:

In order to perform path planning on the aerially obtained LiDAR point clouds, we first generate a 2D obstacle grid for each height layer of the point cloud scan. Combining these scans into a 3D grid gives an accurate assessment of whether an obstacle exists

Methodology (cont.)

within each unit of space. [1] A buffer is applied to every 2D grid using the Breadth First Search (BFS) algorithm, which accounts for UAV size by expanding obstacles outwards. Then, our modified 3D A* algorithm generates a smooth path using this grid and Bezier curves, and then is converted. Namely, it is first translated into point cloud space and then converted into latitude and longitude using the Python *utm* library, which requires the UTM zone of the physical location of the LiDAR scan. This allows our path to be sent to an GPS-equipped UAV. (Fig. 2)

Figure 2: Two cross-sectional 2D grids displaying different heights. [4]

Collision Avoidance:

Avoidance trajectories are calculated at every waypoint in the generated path. The trajectory that meets the distance threshold and is closest to the ending node is selected for each point. If an avoidance is necessary, the UAV will move to the avoidance trajectory and recalculate the best path from that point. Horizontal movement is based on UAV heading (Fig. 3)

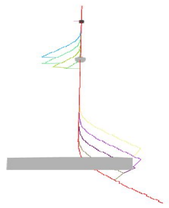

Figure 3: Collision avoidance for two different obstacle types.

Energy Model:

To attempt to increase battery life when UAVs fly the generated path, we have included an energy model that weights nodes differently depending on the direction of travel from the parent node. [2] This model limits energy-demanding directions such as diagonally up whenever possible. (Fig. 4)

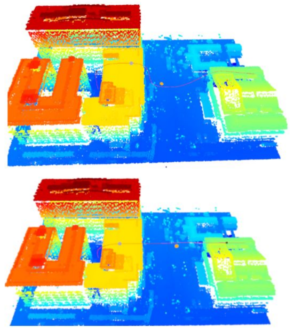

Figure 4: Comparison of path before and after energy model. [3]

API Integration:

We used OpenTopography's API to locate databases in a user-defined search polygon from a generated .json file. Critical information from this .json file is used to remotely navigate OpenTopography's website and download the .las file associated with the search polygon, automating the process.

Results

We have developed a process for path planning using point clouds obtained from USGS 3DEP and can convert the best path into GPS waypoints (Fig. 5). Our modifications successfully account for UAV size and incorporate accurate path smoothing and energy consumption. Our primitive collision detection algorithm works for some basic types of obstacles.

Figure 5: Simulated path in point cloud vs. satellite view using ARDU Pilot. [3]

Results (cont.)

The energy model successfully limits energy intensive movement within the generated path. By using OpenTopography's API and Selenium, we can remotely download point cloud scans from given coordinates. From our testing, we found that 26 nodes w/ path smoothing, and with added sqrt produced the shortest distance without energy function. With the energy function, the fastest path was generally produced with path smoothing and without sqrt and elevation change is limited. Adding the sqrt or energy function increased path generation computation time greatly.

Testing Settings	Distance	A* Computation Time	Elevation Change(m)
26 Nodes (path smoothing, no sqrt)	18868.6986	0.0148	20.7395
10 Nodes (path smoothing, no sqrt)	18974.0368	**0.005**	11.9932
26 Nodes (path smoothing, sqrt)	**18589.6480**	2.8257	**11.779**
10 Nodes (path smoothing, sqrt)	18665.4663	0.9536	11.9952
26 Nodes (no path smoothing, no sqrt)	21426.6782	0.0117	44
10 Nodes (no path smoothing, no sqrt)	20762.2366	0.0178	12
26 Nodes (no path smoothing, sqrt)	19192.2423	2.8640	12
10 Nodes (no path smoothing, sqrt)	19913.7084	0.9038	12

Testing Settings	Distance	A* Computation Time	Elevation Change(m)
26 Nodes (path smoothing, no sqrt)	**19631.779**	**0.9207**	**10**
10 Nodes (path smoothing, no sqrt)	20235.2898	13.1873	**10**
26 Nodes (path smoothing, sqrt)	19644.8975	54.8695	**10**
10 Nodes (path smoothing, sqrt)	20242.1704	14.5067	**10**
26 Nodes (no path smoothing, no sqrt)	20245.7593	1.03158	**10**
10 Nodes (no path smoothing, no sqrt)	20927.9220	14.2147	**10**
26 Nodes (no path smoothing, sqrt)	20245.7593	55.7438	**10**
10 Nodes(no path smoothing, sqrt)	20927.9220	14.1579	**10**

Figure 6: Path statistics without/with energy function.

Conclusion

In conclusion, we were able to create a seamless process for 3D path planning from dual GPS coordinate input. We also were able to test different factors that changed the path generation to find the most optimal path given the criteria we set (length, computational time, and change in elevation). Our process also included a form of primitive collision avoidance for three different types of obstacles (static, dynamic, and vertical objects). This collision avoidance works in real-time and changes the path transmitted to the UAV.

References

[1] Ivan Dryanovski, William Morris, and Jizhong Xiao. "Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles". In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2010, pp. 1553–1559. doi:10.1109/IROS.2010.5652494.

[2] Hasini Viranga Abeywickrama et al. "Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance". In: IEEE Access 6 (2018), pp. 58383–58394. doi: 10.1109/ACCESS.2018.2875040.

[3] USGS 3DEP (2021). WA KingCo 1 2021. U.S. Geological Survey 3D Elevation Program, distributed by OpenTopography.

[4] USGS 3DEP (2021). GA Central 2 2018. U.S. Geological Survey 3D Elevation Program, distributed by OpenTopography.

Acknowledgments

- National Science Foundation – REU Programs
- Department of Defense – Funding
- Auburn University – Program Host