

# UAV Path Planning using Aerially Obtained Point Clouds

Auburn REU on SMART UAVs 2022

Pugh, Alec

University of North Carolina at Chapel Hill  
alecjp@unc.edu

Bower, Luke

The University of Alabama in Huntsville  
lcb0035@uah.edu

Chapman, Richard

Auburn University  
chapmro@auburn.edu

Biaz, Saad

Auburn University  
biazsaa@auburn.edu

June 13, 2022

## Abstract

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 and physical testing using way point transmission show the validity of this method.

## 1 Introduction

The importance of research towards efficient and reliable path finding algorithms has steadily increased with the growing popularity and use of autonomous unmanned aerial vehicles (UAVs). There are many benefits around the development and use of UAV path finding, and each process developed mainly strives to satisfy each of the following requirements: time preservation, path optimality, and danger avoidance. The consequences of these requirements are the most apparent in military and commercial operations: provide a reduction in the cost spent towards training UAV pilots, a reduced amount of human manpower needed to pilot the UAVs, and a reduction of accidents due to human error or incorrect decision making. Technological developments have allowed humans to power cars with batteries for hundreds of miles at a time. While such batteries are viable for larger vehicles, versatile UAVs are compact and cannot carry large batteries or payloads due to their size and thrust output. As such, a typical commercially viable UAV has an average flight time of approximately 15-25 minutes [6]. With flight times this short, being able to accomplish more during a mission without recharging is critical to success. This correlates directly with the time spent

travelling between points of interest and energy consumption of maneuvers, which path finding methods help reduce.

The USGS 3D Elevation Program is a developing program that aims to provide topographical data for the entire United States. Beginning production in 2016, 84% of the nation has topographical data that has been obtained from aircraft LiDAR scans. These LiDAR scans are displayed in the form of a point cloud, which is a set of 3D points (containing X, Y, and Z coordinates, among other types of data) in space relating to their physical GPS location. The data available through USGS 3DEP can be seen as undoubtedly instrumental in providing accurate path finding for UAV operations, especially for locations that may be inaccessible to humans or ordinary UAV flight operation due to dangerous conditions, FAA (**Federal Aviation Administration**) regulations, or various other impractical reasons. Furthermore, if this information is confined to a specific operating environment and is obtained prior to a mission launch, optimal paths can be found with less computational intensity and can be flown without prior mapping.

To this end, the main purpose of this paper hopes to leverage the vast amount of LiDAR point cloud data obtained through USGS 3DEP and show a concise method to provide path finding using this data. In addition, a secondary objective is to present a method of collision avoidance that also leverages the precomputed data, creating an all-inclusive program. This paper is outlined as follows: Section [2] displays the purpose of the research presented. Section [3] presents a literature review for the previous research done in this field. We outline our proposed method for UAV path planning using aerially obtained point clouds in Section [4]. Real-world testing is performed and documented in Section [??]. Our results follow in Section [??] with a brief summary and outline for future work in Section [??].

## 2 Problem Statement and Motivation

The advantages that UAVs withhold over manned aerial vehicles are apparent; however, there are still many limitations with the currently available technology. One main disadvantage comes from limited battery life that UAVs possess, so research into battery improvements for UAVs has been vast. While various methods for increasing battery life has been tested, such as unlimited endurance (laser-beam in-flight recharging, tethering, and swapping), along with alternative fuel sources (such as hydrogen, methanol, and hydrocarbons) [1], few of these methods or alternatives have become common practice for use in commercial and consumer UAVs as of today. This is often due to the limited weight tolerance that UAVs allow, which naturally prevents the trivial solution of equipping a UAV with additional batteries. Therefore, off-board optimization techniques can be viewed as a solution to help optimize the use of the limited battery life available while also preventing weight additions on UAVs with already limited payload tolerances.

As stated previously, obtaining point cloud scans of a given area can be difficult or impossible due to lack of proper equipment, state/governmental regulations, or safety concerns. By taking advantage of the datasets already available, this eliminates the need for expensive equipment and the inaccessibility of obtaining topographical information otherwise. However, the accuracy of path planning using point cloud data is proportional to the quality of a given point cloud scan. With USGS 3DEP datasets, LiDAR scan quality is standardized through QL (quality level) specifications which are detailed enough to be trusted for use for path finding operations. In addition, USGS has plans to continue to increase the overall scan quality overtime through dataset updates [21]. This shows that the accuracy of the methodology presented will increase linearly with the progression of future processes and technologies used to obtain these scans. Therefore, a streamlined 3D path planning approach using this data will be useful for not only present UAV operations, but also the future.

The primary objective of our research is to create a reliable and efficient 3D path that works seamlessly with 3DEP scans. Due to the nature of these scans being obtained through aircraft flying above an area, only the surface points of an environment may be scanned accurately. Thus, features that exist directly perpendicular to a surface (such as building faces underneath a roof, or steep cliffs) may be represented as empty space in a 3DEP scan. The process presented accounts for this problem. Our research uses a multi-rotor UAV for its quick movement and directional freedom. Due to the path being generated off

board from the UAV, on-board equipment can be limited to a 2D LiDAR sensor, GPS, and transmitter. A secondary objective of this research is to have systems in place to provide collision avoidance procedures. By using the precomputed LiDAR data for collision avoidance, we once again do not require additional onboard equipment, which provides greater flexibility for deployment applications. The final objective is to incorporate API integration to create a seamless path generation process from input to output.

### 3 Literary Analysis

There has been many significant contributions to the field of UAV research over the years. To provide a top-down approach in topic discussion, this section will first provide a classification for the different types of mapping systems within UAV research. Then, we provide definitions and related work with UAV and LiDAR, path planning, and collision avoidance techniques with respect to the type of mapping that reflects this paper and process.

#### 3.1 Mapping Classifications

In general, there are three main types of mappings used in autonomous UAV navigation and path planning: map-building, map-less, and map-based systems [13]. *Map-building* systems use a UAV to simultaneously build a map of the environment while flying within it. This is referred to as the "simultaneous localization and mapping" (**SLAM**) computational problem. Work in this area commonly uses onboard sensors (such as LiDAR) to perform both the mapping and localization. Algorithms used for SLAM operate with probabilistic quantities (a dimensionless set of values that build the environment), with two major implementations being the extended Kalman filter (**EKF**) SLAM and GraphSLAM [22]. The benefits of SLAM can be clearly seen for use in environments with unreliable GPS navigation [8], environment reconstruction [19], or as a means to perform autonomous navigation as a whole.

A *map-less* system seeks to perform navigation without using any maps. This is often accomplished using computer vision and reinforcement learning techniques [10], in which information about an environment is constantly being given to a UAV and decisions are made using that data. One method uses a computer vision algorithm to identify the features of obstacles in a process known as *feature tracking*, and the position of a UAV within its environment is calculated based on such features [16]. This method of path planning is great for unknown locations; however, due to the increasing amount of LiDAR data collected from USGS that covers the majority of the United States, there are less computer intensive methods of path planning.

Finally, the *map-based* system, which is the type of system that this paper is based upon, uses a predefined map layout and allows a UAV to navigate based on that information. One way obstacles are defined in a map-based system is through an obstacle grid, which is a 2D grid that stores information on whether a cell is an obstacle or not by giving either a value of 0 or 1, with one representing an obstacle and zero representing free space. Dryanovski *et al.* developed a method for producing multi-volume occupancy grids, which store three-dimensional information about both free and occupied space of an environment and continuously corrects a map when given updated sensor information [4]. However, the focus of this paper is not on pathfinding or collision avoidance, and builds maps from point clouds obtained from ground robots or 3D cameras, as opposed to LiDAR scans from airplanes. Another common method of mapping within map-based systems uses *octrees*. With octrees, a node of space is represented as a 3D cube, called a voxel. Through recursion, each voxel is further subdivided into eight children until a minimum size is obtained, which is predefined and also dictates the resolution of the octree. Because octrees are represented as a tree data structure, previous (lower) resolutions can be easily accessed at any time [9]. However, octrees by nature can only provide approximations of an environment, as voxels are represented as cubes. As such, an approximation of an object with surface curvature (i.e. a sphere or dome) will never be perfectly represented regardless of octree depth.

### 3.2 LiDAR and UAVs

LiDAR is a remote sensing method that uses pulsed lasers to emit light waves towards an environment. The time it takes for these pulses to bounce off of objects in the environment and return to the sensor is measured, and that distance is used to determine the height of the objects. LiDAR points are initially returned as spherical coordinates, but can be converted into cartesian using mathematical formulae, which are displayed below with radius  $\rho$ , polar angle  $\theta$ , and azimuth  $\varphi$ .

$$\begin{aligned}x &= \rho \sin \theta \cos \varphi \\y &= \rho \sin \theta \sin \varphi \\z &= \rho \cos \theta\end{aligned}$$

Alternatively, an onboard Inertial Processing Unit (**IMU**), often equipped on aircraft and used to report the orientation of the vessel, can be used to automatically convert LiDAR data from spherical to cartesian coordinates.

The applications of LiDAR scanners often mesh well with the operational capabilities of UAVs, making their usage together well documented in research. Specifically, there have been many applications resulting from equipping a UAV with a LiDAR scanner such as for mapping, inspection, or measuring purposes. The general process for direct mapping and some inspection applications is to obtain a point cloud via a LiDAR-equipped UAV, and then perform a feature extracting technique to remove noise and enhance the data related to the problem at hand. G. E. Teng *et al.* used this process to inspect the features and security of a power line [20]. Measuring applications follow a similar process: develop post-processing algorithms that interpret the obtained LiDAR data for desired information. Chisholm *et al.* have recently used a LiDAR equipped UAV to estimate the diameter of trees below canopies [3], which shows a promising future for remote sensing using LiDAR in GPS-denied environments. It should be mentioned that most applications with LiDAR and UAVs involve obtaining a scan using the UAV, and then afterwards (or simultaneously in the case of SLAM) performing some processing on the data. This paper focuses on performing UAV flight operations on LiDAR scans obtained via another source (from aircraft).

### 3.3 UAV Path Planning

Path planning techniques all exist to solve the same problem: given an environment with obstacles and a start/goal node, how can a path be constructed between these two nodes while avoiding all obstacles? As described in [13], there are two main types of UAV path planning: global and local. *Global path planning* techniques find a path from the start to the end node within a global (static) map environment. *Local path planning* works to dynamically find subsets of a global path based on UAV state and updated stimuli. It can be seen how global and local path planning are related. A global path remains constant, but can be adjusted with intermediate local paths based on new information obtained from sensors (i.e. appearance of an obstacle, environment that differs from expected, etc.). There are many different algorithms that achieve these goals, with each belonging to a certain paradigm that describes its process at a categorical level [23]. This review will mostly cover research involving UAV pathfinding for node and sample-based algorithms with a focus on real-world environments.

Sampling-based algorithms, such as the Rapidly-exploring Random Tree (**RRT**) and RRT\* algorithms operate by *sampling* points randomly in some  $\mathbb{R}^n$  space. RRT then uses these points as seeds to grow trees from, and connects the grown trees together to form a path between a start and goal point. Many variations have been made to the base RRT algorithm for UAV path planning purposes, such as introducing a bidirectional variant [14] or reducing the search space by growing trees from the start and end points instead of just the initial point [25] to reduce computation time. 3D UAV navigation using sampling-based algorithms and point clouds has also been documented [26], however these algorithms may fail if cavities are present in the point cloud scans.

Node-based algorithms, such as the *A\* algorithm* have been widely used for UAV path planning purposes as well. This is due to its ability to be easily modified to fit the subjects of specific papers [5]. It works by

calculating the total cost  $F(n)$  of each node using two components, a metric for node distance travelled so far  $g(n)$  and a heuristic function  $h(n)$ . The heuristic function chosen can vary the resulting path greatly, and is itself a subject of many papers. By processing the nodes with the lowest cost first using a priority queue, the algorithm prevents unnecessary exploration of too costly nodes, which reduces the computation time. The formula to calculate a node's cost value is shown below.

$$F(n) = g(n) + h(n)$$

The A\* algorithm has been shown to generate almost optimal paths for 3D UAV path planning, with shorter lengths and less computation time compared to the popular sampling-based algorithm, RRT [24].

### 3.4 Collision Avoidance

Collision avoidance is necessary for safe and fully-functional autonomous flight, and the different systems used to accomplish this has undoubtedly contributed to the vast amount of research that has been done in this area. The two key concepts that make up collision avoidance systems are sensing and detection, and maneuver approaches [18]. Sensing and detection, as can be implied from the name, constitute devices or methods that allow a UAV to position itself relative to other obstacles. One technique is optic flow, where motion is detected from brightness patterns of an image and used to calculate an optical flow field, consisting of vectors for each pixel [2, 15, 7]. Stereo-vision techniques use the triangulation of two sensors to obtain depth maps, which is used to perform an avoidance [17]. Laser sensor techniques use sensors such as LiDAR scanners to perform detection and avoidance, and can be seen in [12, 11].

Maneuver approaches are more varied, as the optimality of an approach would depend on the sensing method used and environment that the UAV is operating in. Many different techniques have been developed, and are covered by [18].

## 4 Approach

### 4.1 Software and Equipment Used

Software/Equipment	Reasoning
OpenTopogphy(and API)	We used OpenTopogphy because it is one of the largest combined databases of LiDAR data spanning across 84 percent of the United States and various country's across the world. This website also has an API that allows for the json data for the different databases to be accessed allowing for their LiDAR data to be located and downloaded remotely.
AWS S3	This is Amazons servers that the LiDAR data sets are stored on and can be publicly accessed and downloaded from. They can also be accessed from here via terminal and downloaded remotely.
LasZip	This software running in the background allows for us to be able to convert our .laz files to any type we need including .las and the ASCII version of the .las files.
Open3D	This software allows us to visualise the LiDAR data and see our generated path overlaid on it allowing us to simulate flight and the changes in the path along with testing basic collision avoidance with preset obstacles.
ARDU Pilot	This software allows for communication from a computer to be transmitted to a UAV and a simulated UAV for both real life testing and simulation.
MAVProxy	This is an in-between software that allows for ARDU Pilot and Mission Planner to communicate and link over network IP addresses.
Mission Planner	This is a simulation and transmission software that can control UAVs allowing for different visuals and real life dynamics to be added to the simulation.
LiDAR Scanner	This is equipped to the UAV reading in distances in-front of the UAV and if it finds an unexpected object it will start a collision avoidance sequence.

### 4.2 Process

Our process for computing path planning and collision avoidance for aerially obtained point clouds is outlined below. The process has been subdivided into multiple sections for better comprehension. Each part will discuss the methodology used to accomplish the goal outlined in the title of the section.

#### 4.2.1 Matrix Generation

To generate the 3D matrix  $M$  that is used for path planning, we first must import the .las file that contains the desired point cloud. To do this, we use the *laspy* library in Python. Additionally, we use *laspy* commands to extract the maximum and minimum point cloud points from the data. The difference between the maximum and minimum Z point ( $Z_{max}$  and  $Z_{min}$ ) defines the depth of the matrix. Similarly, the difference between the maximum and minimum Y ( $Y_{max}$  and  $Y_{min}$ ) and X points ( $X_{max}$  and  $X_{min}$ ) define the rows and columns, respectively.

$$M_x = X_{max} - X_{min}$$

$$M_y = Y_{max} - Y_{min}$$

$$M_z = Z_{max} - Z_{min}$$

An elevation grid is then created from the imported .las file. In this 2D grid, the X and Y coordinates represent the relative point cloud location, and the value within that cell represents the height of that section in meters. This is done using code modified from [cite], and will be briefly explained. [explanation]

To convert the 2D grid into a 3D occupancy grid, we loop using height values that start from the minimum Z point to the maximum Z point in the point cloud, creating a temporary 2D grid and assigning a value of 1 or 0 depending on whether or not the cell is an obstacle or not, respectively. This is determined by subtracting the current height value from the height found in the elevation grid at the specified pixel. A threshold value is used to create a *vertical* buffer for object classification within the matrix. The algorithm is displayed as Algorithm [1].

---

**Algorithm 1** create3DMatrix(*height\_values*, *buffer\_size*, *threshold*)

---

```

1:  $M \leftarrow matrix$ 
2:  $height\_values \leftarrow H$ 
3: for  $cur\_height = Z_{min}$  to  $Z_{max}$  do
4:    $G \leftarrow grid$ 
5:   for  $x, y \in G$  do
6:     if  $cur\_height - H[x, y] \leq threshold$  then
7:        $G[x, y] \leftarrow obstacle$ 
8:     else
9:        $G[x, y] \leftarrow free$ 
10:    end if
11:  end for
12:   $G \leftarrow bfs(G, buffer\_size)$ 
13:   $M \leftarrow append(G)$ 
14: end for
15: return  $M$ 

```

---

A *horizontal* buffer is created using an implementation of the Breadth First Search (**BFS**) algorithm. Obstacles are expanded outwards by a predefined amount. The inclusion of the vertical and horizontal buffers were for three main reasons: to help prevent potential collisions due to noise within the point cloud, to account for the size of the UAV itself, and to provide a safe distance between the UAV and the obstacle for path finding and smoothing purposes.



Figure 1: Two cross-sectional 2D grids displaying different heights generated using Algorithm [1].

#### 4.2.2 Path Planning

To develop the path planning portion of this project, we needed something easily modifiable that would work smoothly with the matrix generation technique discussed in the previous section. For these reasons, the A\* path finding algorithm was used as a base, and modified further to suit the needs of this project.

### 4.2.3 Conversions

a

### 4.2.4 Collision Avoidance

a

### 4.2.5 API Integration

With our ultimate goal being to create a seamless process from the input of GPS coordinates to the UAV flying the generated path with collision avoidance it was apparent that we needed to utilize an API. OpenTopogphy has a limited API for developers. This API allows for two GPS coordinates broken down into the maximum and minimum latitude and longitude to create a search polygon. This polygon is then filtered on the biases of metadata, output format, and searching federal databases like USGS. The API then searches with the filtered data to produce a json file giving vital information about the databases that are inside the search area. By passing this information into geoapis naming convention and searching Amazons AWS S3 servers the .laz files in the searched database can be downloaded remotely. The .laz file produced is an calumnation of the point cloud points for the searched area using a web mercator conversion factor EPSG followed by four digits. This factor is extracted from the json file and allows for the point cloud to be converted to GPS coordinates. For most point cloud data sets the EPSG code is 3857 with an x and y conversion factors of:

$$x = \text{radians}(\text{longitude}) * 6378137.0$$
$$y = \log \left( \tan\left(\frac{\pi}{4} + \frac{\text{radians}(\text{latitude})}{2}\right) \right) * 6378137.0$$

The z coordinate is directly related to the mean height above sea level form the NAVD88 standard biased from a tide station in Rimouski, Quebec. These conversion factors convert EPSG 3857(web mercator 84) to EPSG 4326(latitude and longitude) including the height of those points related to the z coordinate. These conversions allow for us to set the starting point of the path generation to the input GPS coordinate saving time from additional input and allowing for the final path to be converted to GPS coordinates once the program has finished computing.

a



## References

- [1] “A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects”. In: *Applied Energy* 255 (2019), p. 113823. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2019.113823>.
- [2] Haiyang Chao, Yu Gu, and Marcello Napolitano. “A survey of optical flow techniques for UAV navigation applications”. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2013, pp. 710–716. DOI: [10.1109/ICUAS.2013.6564752](https://doi.org/10.1109/ICUAS.2013.6564752).
- [3] Ryan A. Chisholm et al. “UAV LiDAR for below-canopy forest surveys”. In: *Journal of Unmanned Vehicle Systems* 01.01 (2013), pp. 61–68. DOI: [10.1139/juvs-2013-0017](https://doi.org/10.1139/juvs-2013-0017). eprint: <https://doi.org/10.1139/juvs-2013-0017>. URL: <https://doi.org/10.1139/juvs-2013-0017>.
- [4] 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](https://doi.org/10.1109/IRoS.2010.5652494).
- [5] Daniel Foead et al. “A Systematic Literature Review of A\* Pathfinding”. In: *Procedia Computer Science* 179 (2021). 5th International Conference on Computer Science and Computational Intelligence 2020, pp. 507–514. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.01.034>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921000399>.
- [6] Boris Galkin, Jacek Kibilda, and Luiz A. DaSilva. “UAVs as Mobile Infrastructure: Addressing Battery Lifetime”. In: *IEEE Communications Magazine* 57.6 (2019), pp. 132–137. DOI: [10.1109/MCOM.2019.1800545](https://doi.org/10.1109/MCOM.2019.1800545).
- [7] Zdzislaw Gosiewski, Jakub Ciesluk, and Leszek Ambroziak. “Vision-based obstacle avoidance for unmanned aerial vehicles”. In: *2011 4th International Congress on Image and Signal Processing*. Vol. 4. 2011, pp. 2020–2025. DOI: [10.1109/CISP.2011.6100621](https://doi.org/10.1109/CISP.2011.6100621).
- [8] Sebastian Hening et al. “3D LiDAR SLAM Integration with GPS/INS for UAVs in Urban GPS-Degraded Environments”. In: *AIAA Information Systems-AIAA Infotech @ Aerospace*. DOI: [10.2514/6.2017-0448](https://doi.org/10.2514/6.2017-0448). eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2017-0448>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2017-0448>.
- [9] Armin Hornung et al. “OctoMap: an efficient probabilistic 3D mapping framework based on octrees”. In: *Autonomous Robots* 34 (2013), pp. 189–206.
- [10] Sungwoo Jung and David Hyunchul Shim. *Mapless Navigation: Learning UAVs Motion for Exploration of Unknown Environments*. 2021. DOI: [10.48550/ARXIV.2110.01747](https://doi.org/10.48550/ARXIV.2110.01747). URL: <https://arxiv.org/abs/2110.01747>.
- [11] Cezary Kownacki. “A concept of laser scanner designed to realize 3D obstacle avoidance for a fixed-wing UAV”. In: *Robotica* 34.2 (2016), pp. 243–257.
- [12] Liang Lu et al. “Laser-based Collision Avoidance and Reactive Navigation using RRT and Signed Distance Field for Multirotor UAVs”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2019, pp. 1209–1217. DOI: [10.1109/ICUAS.2019.8798124](https://doi.org/10.1109/ICUAS.2019.8798124).
- [13] Yuncheng Lu et al. “A survey on vision-based UAV navigation”. In: *Geo-spatial Information Science* 21.1 (2018), pp. 21–32. DOI: [10.1080/10095020.2017.1420509](https://doi.org/10.1080/10095020.2017.1420509). eprint: <https://doi.org/10.1080/10095020.2017.1420509>. URL: <https://doi.org/10.1080/10095020.2017.1420509>.
- [14] Li Meng, Song Qing, and Zhao Qin Jun. “UAV path re-planning based on improved bidirectional RRT algorithm in dynamic environment”. In: *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. 2017, pp. 658–661. DOI: [10.1109/ICCAR.2017.7942779](https://doi.org/10.1109/ICCAR.2017.7942779).
- [15] Paul Clark Merrell, Dah-Jye Lee, and Randal W Beard. “Obstacle avoidance for unmanned air vehicles using optical flow probability distributions”. In: *Mobile Robots XVII*. Vol. 5609. International Society for Optics and Photonics. 2004, pp. 13–22.

- [16] Ivan Fernando Mondragon et al. “Visual Model Feature Tracking For UAV Control”. In: *2007 IEEE International Symposium on Intelligent Signal Processing*. 2007, pp. 1–6. DOI: [10.1109/WISP.2007.4447629](https://doi.org/10.1109/WISP.2007.4447629).
- [17] Jongho Park and Youdan Kim. “Stereo vision based collision avoidance of quadrotor UAV”. In: *2012 12th International Conference on Control, Automation and Systems*. 2012, pp. 173–178.
- [18] Hung Pham et al. “A survey on unmanned aerial vehicle collision avoidance systems”. In: *CoRR* abs/1508.07723 (2015). arXiv: [1508.07723](https://arxiv.org/abs/1508.07723). URL: <http://arxiv.org/abs/1508.07723>.
- [19] Zhexiong Shang and Zhigang Shen. “Real-time 3D Reconstruction on Construction Site using Visual SLAM and UAV”. In: *CoRR* abs/1712.07122 (2017). arXiv: [1712.07122](https://arxiv.org/abs/1712.07122). URL: <http://arxiv.org/abs/1712.07122>.
- [20] GE Teng et al. “Mini-UAV LiDAR for power line inspection”. In: *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* (2017), pp. 297–300.
- [21] *USGS 3DEP Topographic Data Quality Levels (QLs)*. URL: <https://www.usgs.gov/3d-elevation-program/topographic-data-quality-levels-qls>.
- [22] Wikipedia. *Simultaneous localization and mapping* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Simultaneous%20localization%20and%20mapping&oldid=1092128572>. [Online; accessed 08-June-2022]. 2022.
- [23] Liang Yang et al. “A literature review of UAV 3D path planning”. In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*. 2014, pp. 2376–2381. DOI: [10.1109/WCICA.2014.7053093](https://doi.org/10.1109/WCICA.2014.7053093).
- [24] Christian Zammit and Erik-Jan Van Kampen. “Comparison between A\* and RRT algorithms for UAV path planning”. In: *2018 AIAA guidance, navigation, and control conference*. 2018, p. 1846.
- [25] Denggui Zhang, Yong Xu, and Xingting Yao. “An Improved Path Planning Algorithm for Unmanned Aerial Vehicle Based on RRT-Connect”. In: *2018 37th Chinese Control Conference (CCC)*. 2018, pp. 4854–4858. DOI: [10.23919/ChiCC.2018.8483405](https://doi.org/10.23919/ChiCC.2018.8483405).
- [26] Zhaoliang Zheng, Thomas R. Bewley, and Falko Kuester. “Point Cloud-Based Target-Oriented 3D Path Planning for UAVs”. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2020, pp. 790–798. DOI: [10.1109/ICUAS48674.2020.9213894](https://doi.org/10.1109/ICUAS48674.2020.9213894).