

Article

Path Planning for Unmanned Aerial Vehicles in Complex Environments

César Gómez Arnaldo *, María Zamarreño Suárez , Francisco Pérez Moreno and Raquel Delgado-Aguilera Jurado 

Department of Aerospace Systems, Air Transport and Airports, Universidad Politécnica de Madrid (UPM), 28040 Madrid, Spain; maria.zamsuarez@upm.es (M.Z.S.); francisco.perez.moreno@upm.es (F.P.M.); raquel.djurado@upm.es (R.D.-A.J.)

* Correspondence: cesar.gomez.arnaldo@alumnos.upm.es

Abstract: This paper introduces a comprehensive framework for generating obstacle-free flight paths for unmanned aerial vehicles (UAVs) in intricate 3D environments. The system leverages the Rapidly Exploring Random Tree (RRT) algorithm to design trajectories that effectively avoid collisions with structures of diverse shapes and sizes. Discussion revolves around the challenges encountered during development and the successful achievement of generating collision-free routes. While the system represents an initial iteration, it serves as a foundation for future projects aiming to refine and expand upon its capabilities. Future work includes simulation testing and integration into UAV missions for image acquisition and structure scanning. Additionally, considerations for swarm deployment and 3D reconstruction using various sensor combinations are outlined. This research contributes to the advancement of autonomous UAV navigation in real-world scenarios.

Keywords: unmanned aerial vehicles (UAVs); rapidly exploring random tree (RRT) algorithm; obstacle avoidance; 3D environment; collision-free trajectories; autonomous navigation



Citation: Gómez Arnaldo, C.; Zamarreño Suárez, M.; Pérez Moreno, F.; Delgado-Aguilera Jurado, R. Path Planning for Unmanned Aerial Vehicles in Complex Environments. *Drones* **2024**, *8*, 288. <https://doi.org/10.3390/drones8070288>

Academic Editor: Yushu Yu

Received: 30 May 2024

Revised: 19 June 2024

Accepted: 24 June 2024

Published: 26 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The implementation of a solution enabling the autonomous scanning and completion of the 3D reconstructions of buildings with drones represents a significant endeavor within contemporary technological landscapes. This research addresses the need for automated systems capable of executing such tasks without direct human intervention, focusing on methodologies to meticulously plan optimal flight routes around structures to capture multi-angle imagery while ensuring safety and collision avoidance.

1.1. Motivation

The rapid development in 3D environments and virtual reality technologies has ushered in a new era of immersive experiences and realistic simulations. This shift has heightened the demand for solutions adept at generating intricate and precise three-dimensional models. Drones have emerged as indispensable tools in this domain, offering unparalleled versatility and efficacy in data capture, thereby enabling nuanced visualizations and a deeper contextual understanding of the captured environments.

The utilization of drones for the capture of three-dimensional landscapes presents a range of practical use cases. In sectors such as tourism and travel, drones facilitate aerial tours and the capture of panoramic views, providing users with novel avenues for virtual exploration and discovery.

In infrastructure inspection, drones ensure safe and efficient access to areas that are otherwise inaccessible or hazardous for human operatives [1,2]. In filmmaking and documentary production, drones have revolutionized aerial cinematography, offering unprecedented perspectives and enhancing storytelling capabilities.

Central to the development of this solution is the challenge of orchestrating optimal flight paths around diverse structures to capture imagery from varied vantage points while

mitigating the risk of collisions. Ensuring the safety of equipment and preserving environmental integrity are crucial considerations guiding the design of the autonomous system.

This research seeks to devise an autonomous framework enabling drones to undertake the scanning and reconstruction of buildings and structures. The overarching objective is to achieve high precision in data capture while promoting the adoption of cutting-edge technologies across diverse domains.

The findings from this research have significant implications within the realms of drone technology and three-dimensional environment creation, potentially revolutionizing processes across various sectors and enhancing efficiency, precision, and safety in 3D data acquisition.

1.2. Objectives

This research aims to integrate and advance upon previous studies [3–6] to develop an autonomous path-planning algorithm [7] for unmanned aerial vehicles (UAVs) in 3D scene reconstruction [8]. The primary objective is to devise a collision-free path-planning algorithm enabling UAVs to execute reconnaissance missions around specific scenes.

In this initial approach, our focus is on defining the essential requirements for capturing high-quality images of specific objects. The current path-planning algorithm is designed to meet these requirements without incorporating time-related or cost-related metrics. This means that parameters such as total execution time and mission completion cost are not yet considered in the simulations. The primary objective is to ensure collision-free navigation and optimal image capture.

Future work will involve the inclusion of metrics like mission completion cost, total execution time, and collision risk thresholds. This will allow us to develop a robust model that not only achieves the desired image quality but also optimizes for cost efficiency and timely mission completion.

Furthermore, in this review of UAV path-planning techniques, we focus on methods suited for static obstacle environments [9]. Significant advancements have been made in addressing dynamic and complex scenarios. For example, studies on drone–rider joint delivery mode [10], hybrid beetle swarm optimization [11], and multi-objective location-routing problems in post-disaster stages [12] provide valuable insights into routing optimization and dynamic obstacle management. These studies highlight advanced techniques for handling real-time changes and dynamic obstacles in the environment. Despite these advancements, our framework specifically addresses static environments to ensure collision-free navigation around static structures. This foundational approach aims to establish robust path-planning methodologies that can later be extended to dynamic scenarios.

The secondary objective is to explore and implement state-of-the-art technologies in object recognition and image segmentation [13,14]. Specifically, we aim to incorporate SAM technology (Segment Anything Model [15]) to segment and select objects within images, facilitating collision-free route planning.

In conjunction with the objectives, this research aims to unify previous developments and technologies to enhance system capabilities. This involves two primary development branches:

- **Scene information acquisition through orthophotographs:** This entails extracting orthophotographs from cities and other surfaces using multiple drones to obtain aerial images [16,17]. The goal is to apply this system to extract the point clouds of buildings for path planning. The resulting system will be based on previous successes in orthophotograph generation.
- **3D Reconstruction:** The objective is to obtain data necessary for 3D reconstructions as close to reality as possible [18]. This involves generating routes for UAVs to capture the comprehensive imagery of structures, ensuring complete reconstructions without gaps. The system will build upon previous work to guide the process and validate route selection and sensor characteristics.

The final objective is to employ various methods and techniques for information acquisition during missions to improve reconstruction quality. Tests with different sensors [19] will be conducted, comparing their performance and suitability for the path-planning algorithm. Additionally, improvements in LiDAR systems, such as implementing a 3D LiDAR [20], will be explored.

The goal of this research is to document the techniques applied to information extraction, object reconstruction, and route definition for collision-free autonomous flights. This documentation will include an analysis of applied concepts to highlight those yielding the best results.

This research offers the opportunity to acquire cross-disciplinary knowledge applicable to various fields, enriching general understanding and contributing to advancements in computer science, robotics, aeronautics, and related disciplines.

2. State of the Art

Three-dimensional reconstruction using autonomous drones has emerged as a key discipline across various fields, driven by technological advancements in perception and navigation. In cartography, archaeology, precision agriculture, and construction, drones offer the efficient capture of three-dimensional data. In archaeology, they facilitate cultural heritage preservation and the understanding of past civilizations. In agriculture, they enhance crop and resource management. In construction, they streamline project supervision. Additionally, they have applications in disaster management, surveillance, mining, and environmental management. In summary, this versatile technology promises positive impacts on decision making and research in diverse sectors, enabling the better understanding and utilization of the environment. Its ongoing evolution holds even greater benefits for society.

2.1. Mapping and Perception Techniques for Autonomous Drones

Autonomous drones rely on a variety of techniques and algorithms to generate detailed maps and perceive the environment in 3D. These techniques play a fundamental role in the data capture process and in the creation of precise and up-to-date three-dimensional models. Below, we explore some of the most important techniques used by autonomous drones to perform these tasks:

1. **SLAM (Simultaneous Localization and Mapping):** SLAM is a key technique for autonomous drones that allows them to build maps of the environment while simultaneously estimating their position within that environment. Through SLAM, drones can navigate and explore unknown areas using sensor information (such as cameras, IMU, and LiDAR) to determine their position and create a real-time map. This technique is essential for autonomous flights and for operating in GPS-denied environments, such as indoors or in areas with poor satellite coverage.
2. **Photogrammetry:** Photogrammetry is a technique that involves capturing and analyzing images from different angles to reconstruct objects and environments in 3D. Autonomous drones equipped with cameras can take aerial photographs of an area and then, using point matching and triangulation techniques, generate accurate three-dimensional models of the terrain surface or structures in the mapped area [21].
3. **LiDAR (Light Detection and Ranging):** LiDAR is a sensor that emits laser light pulses and measures the time it takes to reflect off nearby objects or surfaces. By combining multiple distance measurements, a 3D point cloud representing the surrounding environment is obtained. Autonomous drones equipped with LiDAR can obtain precise and detailed data on the terrain, buildings, or other objects, even in areas with dense vegetation or complex terrain.
4. **Multi-sensor data fusion:** Autonomous drones may be equipped with multiple sensors, such as RGB cameras, thermal cameras, LiDAR, and IMU. Multi-sensor data fusion allows combining information obtained from different sources to obtain a more complete and detailed view of the environment. The combination of visual and depth

data, for example, improves the accuracy and resolution of the 3D models generated by drones [22].

5. **Object recognition and tracking:** Autonomous drones can use computer vision techniques and machine learning to recognize the objects of interest in real-time and track their position and movement. This capability is useful in applications such as industrial inspections, surveillance, and the mapping of urban areas [23].
6. **Route planning and intelligent exploration:** Route planning algorithms allow autonomous drones to determine the best trajectory to cover an area of interest and collect data efficiently. Intelligent route planning is essential to maximize the coverage of the mapped area and optimize flight duration.

2.2. Image Segmentation Techniques and Object Recognition

The application of SAM [24], the artificial intelligence model developed by Meta's research department (Facebook's parent company), represents a significant advancement in the field of visual perception and object recognition in images and videos. This technology has the potential to play a crucial role in 3D reconstruction with autonomous drones, providing substantial improvements in identifying and capturing data for generating detailed and accurate 3D models.

SAM is based on advanced artificial intelligence techniques, specifically convolutional neural networks (CNNs) [25] and deep learning. Through training on large amounts of visual data, SAM models have the ability to recognize and classify a wide variety of objects, including those that were not part of its training set. This generalization feature is particularly relevant in 3D reconstruction with drones, where the ability to identify objects and structures in real-time is essential for planning optimal routes and capturing precise data.

One of the key benefits of SAM in 3D reconstruction with autonomous drones is its capability to detect and recognize objects in real-time, facilitating more efficient and precise data capture. During drone flights, SAM can identify architectural elements, facade details, vehicles, people, and other objects of interest within the captured images or videos. This valuable information enhances route planning and viewpoint selection for data capture, ensuring comprehensive scene coverage and minimizing the risk of potential collisions. Integrating SAM into the process of 3D reconstruction with drones also has a positive impact on the quality and fidelity of the generated models [26]. By correctly identifying and classifying objects present in the images, SAM can contribute to greater accuracy in reconstructing the three-dimensional environments captured. Detecting details and structural elements on building facades, for example, allows for a more realistic and comprehensive representation of the scene.

Additionally, SAM can be used as an assistive tool in the post-processing phase of captured data. By identifying specific objects and elements in the images, it facilitates the automatic segmentation and classification of data, simplifying the work of experts in processing and analyzing the captured information.

However, it is important to consider some challenges associated with implementing SAM in 3D reconstruction with autonomous drones. One of them is the need for adequate connectivity and processing power to run the model in real-time on board the drone. Transferring large amounts of data over the network and real-time processing require robust and efficient infrastructures.

2.3. Three-Dimensional Reconstruction Methods

Three-dimensional reconstruction from the data collected by drones is a crucial task for converting real-world captured information into precise and detailed three-dimensional digital models. There are several techniques and approaches to achieve this task, each with its advantages and limitations. Below, we explore in detail some of the most-used methods for 3D reconstruction with drones:

1. **Point cloud and mesh methods:** Point cloud is a common technique in 3D reconstruction where the data captured by sensors, such as LiDAR or RGB cameras, are represented as a collection of points in a 3D space. These points can be densely distributed and contain information about the geometry and position of the objects in the environment. Various methods can be applied to generate a three-dimensional mesh from the point cloud, such as Delaunay triangulation [27], Marching Cubes [28], or Poisson Surface Reconstruction [29]. The resulting mesh provides a continuous surface that is easier to visualize and analyze [30].
2. **Fusion of data from different sensors:** Autonomous drones can be equipped with multiple sensors, such as RGB cameras, thermal cameras, and LiDAR. Data fusion from the different sensors allows combining information obtained from each source to obtain a more complete and detailed view of the environment. For example, combining visual data with LiDAR depth data can improve the accuracy and resolution of the generated 3D model. Data fusion is also useful for obtaining complementary information about properties such as the texture, temperature, and reflectivity of the environment.
3. **Volumetric and octrees [31]:** Volumetry is a technique that represents objects and scenes in 3D using discrete volumes instead of continuous surfaces. Octrees are a common hierarchical data structure used in volumetry, which divides space into octants to represent different levels of detail. This technique is especially useful for the efficient representation of complex objects or scenes with fine details.
4. **Machine Learning methods:** Machine learning and artificial intelligence have been increasingly applied in 3D reconstruction with drones. Machine learning algorithms can improve the accuracy of point matching in photogrammetry, object recognition, and point cloud segmentation. They have also been used in the classification and labeling of 3D data to improve the interpretation and understanding of the generated models.

Each of these methods offers distinct advantages and challenges, and the selection of an approach depends on the context and specific objectives of 3D reconstruction. Combining multiple techniques and fusing data from various sources hold promise for achieving more accurate and comprehensive 3D models. The ongoing advancements in technology and research in this field will continue to significantly enhance the quality and application of 3D reconstruction with autonomous drones.

2.4. Relevant Research and Projects

Over the past years, 3D reconstruction with autonomous drones has been the subject of numerous studies and research projects that have demonstrated its potential in a wide range of applications. Below are the summaries of some of the most significant projects and research works related to this technology:

1. **Project: “Drones for archaeological documentation and mapping” [32]:** This project focused on using autonomous drones equipped with RGB cameras for archaeological documentation and mapping. Autonomous flights were conducted over archaeological sites, capturing aerial images from different perspectives. Photogrammetry was used to generate the detailed 3D models of historical structures. The results highlighted the efficiency and accuracy of the technique, allowing the rapid and non-invasive documentation of archaeological sites, facilitating the preservation of cultural heritage.
2. **Research: “Simultaneous localization and mapping for UAVs in GPS-denied environments” [33]:** This research addressed the challenge of autonomous drone navigation in GPS-denied environments, such as indoors or densely populated urban areas. A specific SLAM algorithm for drones was implemented, allowing real-time map creation and position estimation using vision and LiDAR sensors. The results demonstrated the feasibility of precise and safe autonomous flights in GPS-denied scenarios, significantly expanding the possibilities of drone applications in complex urban environments.

3. **Project: “Precision agriculture using autonomous drones” [34]:** This project focused on the application of autonomous drones for precision agriculture. Drones equipped with multispectral cameras and LiDAR were used to inspect crops and collect data on plant health, soil moisture, and nutrient distribution. The generated 3D models provided valuable information for agricultural decision making, such as optimizing irrigation, early disease detection, and efficient resource management.
4. **Research: “Deep Learning approaches for 3D reconstruction from drone data” [35]:** This research explored the application of deep learning techniques in 3D reconstruction from the data captured by drones. Convolutional neural networks were implemented to improve the quality and accuracy of the 3D models generated by photogrammetry and LiDAR. The results showed significant advances in the efficiency and accuracy of 3D reconstruction, opening new opportunities for the use of drones in applications requiring high resolutions and fine details.

3. Methodology

This research adopts a systematic approach to address the problem of generating obstacle-free flight paths for unmanned aerial vehicles (UAVs) in complex 3D environments. The methodology encompasses several key stages:

1. **Problem framing:** The research begins with a comprehensive analysis of the problem space, delineating the requirements and constraints associated with UAV path planning in dynamic and cluttered environments. This step involves identifying the primary objectives of the study, such as ensuring collision-free navigation and optimizing flight efficiency.
2. **Literature review:** A thorough review of the existing literature is conducted to gain insights into the state-of-the-art techniques and algorithms for UAV path planning and obstacle avoidance. Emphasis is placed on understanding the principles underlying the Rapidly Exploring Random Tree (RRT) algorithm, a widely used approach in autonomous navigation systems.
3. **Algorithm adaptation:** Building upon the insights gained from the literature review, the RRT algorithm is adapted and tailored to suit the specific requirements of the target application. This involves modifying the algorithm to enable trajectory generation in 3D environments while accounting for obstacles and dynamic spatial constraints.
4. **Software implementation:** The adapted RRT algorithm is implemented in software, leveraging appropriate programming languages and libraries. The implementation process encompasses the development of algorithms for path generation, collision detection, and trajectory optimization, along with the integration of the necessary data structures and algorithms.
5. **Simulation environment setup:** A simulation environment is set up to facilitate the rigorous testing and evaluation of the proposed path-planning system. This involves creating a virtual 3D environment that replicates a real-world scenario.
6. **Performance analysis:** The results obtained from the validation experiments are analyzed and interpreted to evaluate the system’s performance in terms of path quality, collision avoidance capability, computational efficiency, and scalability.
7. **Discussion and reflection:** The findings of the research are discussed in detail, highlighting the strengths, limitations, and potential areas for the improvement of the proposed methodology. Insights gained from the analysis are used to refine and iterate upon the developed system.

This comprehensive methodology ensures a systematic and rigorous approach to addressing the research objectives while facilitating the development of effective solutions for UAV path planning in complex environments.

4. Solution Design

This section outlines the logic, methodology, and architecture devised for creating a system facilitating collision-free route generation from orthophotos. The system archi-

texture comprises three interconnected components aimed at producing a route enabling drones to navigate without colliding with objects during reconstruction missions (Figure 1).

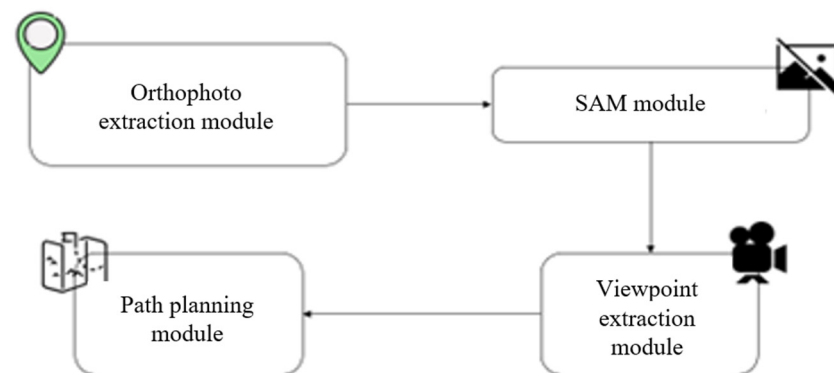


Figure 1. System components.

Building upon prior research, the system leverages the orthophotos and point clouds obtained from scenarios featuring distinct buildings. These serve as targets for the final reconstruction study, for which comprehensive support is intended through the devised process.

The orthophotos derived are subjected to analysis using a segmentation model, facilitating the selection of various objects present in the image. Subsequently, through decomposing the image into specific building contours, points in the coordinate space are identified. These points enable the capture of photographs from various angles, contributing to the comprehensive scanning of the buildings.

Upon pinpointing these coordinates, a collision-free route is generated, ensuring the avoidance of both building structures and extraneous obstacles encountered along the flight path.

4.1. Orthophoto Mission

Orthophotos are chosen as the foundation for the project due to their inherent advantages over the conventional aerial imagery obtained during drone flights:

- Enhanced geographical precision and uniform scale.
- Detailed visual representation.
- Compatibility with map overlay and integration.

Hence, the aim is to build upon previous efforts and continually refine the system's design as it integrates into novel solutions.

The system integrates various components and software to simulate missions where a fleet of drones traverses a predefined area. The drones divide the area into segments, each executing distinct flight patterns to capture diverse perspectives through aerial photography.

Post-simulation, the images captured by drones undergo processing to convert them into orthophotos and point clouds, essential for the subsequent stages of the process.

To facilitate this, a Docker virtual machine instance is deployed, enabling the execution of the *OpenDroneMap* library. This open-source toolkit, available on GitHub, facilitates image processing and conversion into maps and detailed 3D models using photogrammetry techniques.

Critical to the process is obtaining both an orthophoto (.tif file) and a point cloud (.las or .laz file) for subsequent segmentation and analysis.

Furthermore, to validate results and streamline model visualization, the CloudCompare "Unified" v2.13.1 software is considered. This tool allows the representation and conversion of models, ensuring compatibility with subsequent processing steps. The point cloud models in .las format are converted to .ply for seamless integration with the Open3D library in Python, the standard for point cloud manipulation in the language.

4.2. Image Segmentation Model

This module of the solution incorporates image segmentation principles to create a user-interactive final solution. Users can select specific regions of an image for scanning through the viewpoints and routes generated in subsequent sections.

Image segmentation is a fundamental technique in computer vision that aims to divide images into regions corresponding to distinct objects or semantic categories. It is widely used in object detection, scene understanding, video editing, and video analysis.

In recent times, with the rapid advancement of artificial intelligence-based applications, numerous companies have made significant strides in image and video processing. Meta, in particular, has developed a system named SAM (Segment Anything Model), leveraging artificial intelligence trained on vast datasets to achieve high-quality and reliable image and video segmentation.

SAM, trained on millions of input data and masks, autonomously segments objects without requiring domain-specific training or additional supervision. It excels in segmenting faces, clothing, or accessories directly without prior knowledge. Moreover, SAM accepts input prompts such as markers, bounding boxes grouping specific elements in the photograph, or even text indicating the names of the objects present in the image being processed.

Given the magnitude of the developed system, it presents a robust tool for designing a solution enabling user interaction. This includes segmenting specific buildings within a neighborhood featuring multiple residences.

We propose employing this model to fulfill the image segmentation module, adapting it to the project's needs: processing orthophotos and segmenting structures and objects within images. During the post-segmentation of the orthophoto obtained from the preceding module, maintaining the geolocation information of the segmented structure is imperative. This facilitates merging the terrain's point cloud data with segmentation data, allowing the targeted processing of the chosen object.

Viewpoint Selection

The objective of this research is to design a program capable of generating a flight path around a building or object, allowing a drone to conduct reconnaissance and subsequently reconstruct it in 3D scenarios. To achieve this, the route incorporates points from which the drone can capture detailed images, facilitating the creation of a 3D model with high fidelity and robustness relative to real-world qualities [36].

These points are vital in defining the drone's route, influencing the paths it can traverse, and must adhere to multiple constraints.

Primarily, establishing a minimum distance from the structure is crucial to ensure the following:

- Aircraft safety and integrity [37]: A minimum safety distance must be defined to prevent collisions with the building structure. Although regulations do not specify an exact distance, common sense dictates defining this metric.
- Comprehensive structure inspection and surface detail acquisition: Capturing images too close to the building yields detailed but information-intensive photographs. However, closer images require a greater number to cover the entire object volume efficiently.

Among these aspects, the latter carries greater weight, as the safety distance falls within the correct distance range for efficient photography. Therefore, much of the study surrounding viewpoint identification focuses on ensuring no part of the building is omitted from the photographs.

In the simulation environment used in the previous years and intended for this solution's design, various parameters of the cameras and sensors aboard the drone are crucial for estimating the distance at which the aircraft should be positioned when capturing photographs. Considering initial missions employing photographic cameras, the focal length and image resolution parameters are of interest. For simulations, the camera resolutions

are set to 1920×1080 pixels, with a camera field of view (FOV) of 120° . These camera characteristics are pivotal in defining the minimum distance for photography (see Figure 2).

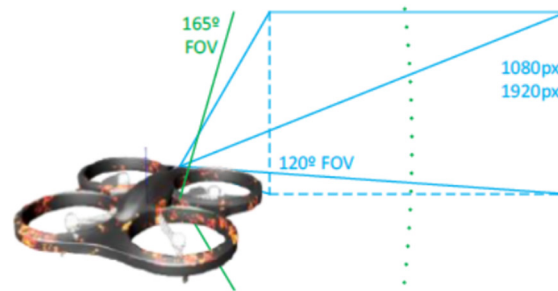


Figure 2. Camera configuration.

Once the field of view has been defined, the horizontal distance that the drone can cover with each image can be calculated, and from the horizontal distance and the resolution of the image, the vertical distance over the surface of the building. To do this, we apply basic geometry concepts as follows such as those shown in Figure 3:

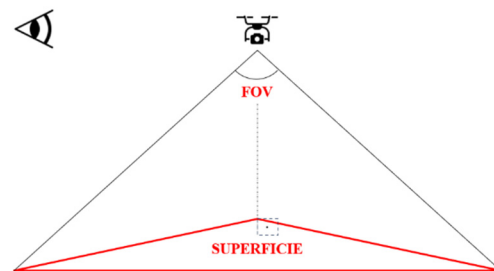


Figure 3. Calculations on the surface.

In this way, the distance to the structure can be defined by choosing the horizontal measure covered by each image, or the horizontal measure can be defined by choosing a specific minimum distance from which the photographs must be taken. During the process of implementing the solution and profiling through testing, multiple checks are performed to specify the distance to the structure from which to take the images.

It is important to note that the aircraft may not always be able to be placed parallel to the surface of the structure, and sometimes the images must be captured at other angles of inclination or incidence on the building (see Figure 4).

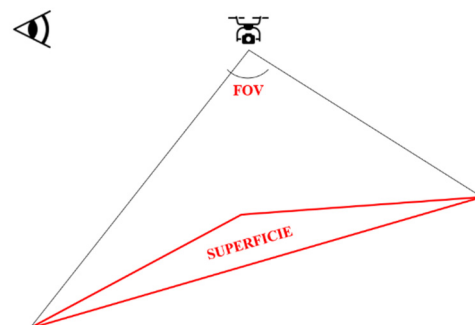


Figure 4. Case of inclined surface.

In these cases, the surface collected is greater than in the example where the drone is parallel to the surface, but since the image is not captured facing the surface, it is possible that other qualities of the building may be captured.

When reconstructing a building in 3D from images, it is beneficial to use photographs of the same surface taken from various perspectives (Figure 5). This approach captures a higher level of detail, such as the depth of wall protrusions, double bottoms in structures, or areas that are obscured from some angles but visible from others. Therefore, it is advantageous to capture images from multiple angles, and having overlapping images is not an issue. This redundancy ensures that all the aspects of the building are thoroughly documented, enhancing the overall accuracy and detail of the 3D reconstruction.

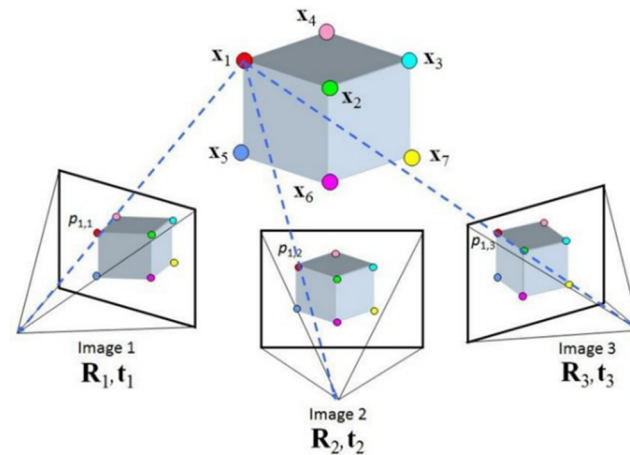


Figure 5. Capturing perspectives of the same point [5].

Another aspect to consider when generating viewpoints, apart from the distance to the structure, is the criterion or method proposed for placing a viewpoint in a specific location. They cannot be placed randomly or arbitrarily one by one because the process can be different and tedious for each structure if performed that way. When representing figures, modeling structures, and shaping objects in a simplified manner, the use of triangles is the best alternative due to the different qualities they present:

- **Simplicity and efficiency.** They are the simplest and most basic geometric shapes, which facilitates their manipulation and mathematical calculations. In addition, graphic engines and rendering algorithms are optimized to work efficiently with triangles.
- **Versatility.** They are flat polygons that can be used to approximate any three-dimensional surface with sufficient detail.
- **Interpolation and smoothing.** They allow the simple interpolation of vertex values, such as textures, colors, or surface normals. Additionally, when enough triangles are used, the resulting surface appears smooth.
- **Lighting calculations.** Triangles are ideal for performing lighting and shading calculations since their geometric and flat properties allow the easy determination of normals and light behavior on the surface.
- **Ease of subdivision.** If more detail is needed on a surface, triangles can be subdivided into smaller triangles without losing coherence in the representation.

Numerous studies propose generating viewpoints around a building by simplifying the structure in this way to work with a triangle mesh and be able to use its vertices and normal vectors to calculate the best coordinates to place viewpoints.

In this case, and after reviewing several approaches focused on solving the aforementioned problem [38,39], a procedure based on the study “Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures” [40] is proposed, simplifying the initial structure into triangles and using their normal vectors to preliminarily place a broad set of viewpoints in space. Consistent with the aforementioned, the drone’s flight path should not contain too many points to avoid becoming too extensive and increasing the total mission cost. Therefore, different transformations will be applied

to the initial viewpoints to group them, reducing the total flight distance of the vehicle but without losing information from the object during scanning.

4.3. Generation of Routes for Unmanned Aerial Vehicles in a 3D Scenario

Route generation is an essential stage in the development of an autonomous unmanned aerial vehicle to complete reconnaissance missions in a 3D scenario. The aim of this stage is to enable the drone to plan safe and efficient trajectories, avoiding collisions with the objects in the environment. To achieve this goal, various path-planning algorithms have been studied, considering their suitability for complex and unknown environments.

4.3.1. Study of Path-Planning Algorithms

In the process of selecting the most suitable algorithm for drone route generation, the following algorithms have been considered:

- **A* (A-Star):** The A* algorithm is widely used in pathfinding problems on graphs and meshes. A* performs an informed search using a heuristic function to estimate the cost of the remaining path to the goal. While A* can be efficient in environments with precise and known information, its performance can significantly degrade in complex and unknown 3D environments due to a lack of adequate information about the structure of the space.
- **PRM (Probabilistic Roadmap):** PRM is a probabilistic algorithm that creates a network of valid paths through the random sampling of the search space. Although PRM can generate valid trajectories, its effectiveness is affected by the density of the search space and may require a high number of sampling points to represent accurate trajectories in a 3D environment with complex obstacles.
- **D* (D-Star):** D* is a real-time search algorithm that recalculates the route when changes occur in the environment. While D* is suitable for dynamic environments, its computational complexity can be high, especially in 3D scenarios with many moving objects and obstacles.
- **Theta* (Theta-Star):** Theta* is an improvement of A* that performs a search in the discretized search space using linear interpolation to smooth the path. While Theta* can produce more direct and efficient trajectories than A*, its performance may degrade in environments with multiple obstacles and complex structures.
- **RRT (Rapidly Exploring Random Tree):** The RRT algorithm is a widely used path-planning technique in complex and unknown environments. RRT uses random sampling to build a search tree that represents the possible trajectories of the UAV. Its probabilistic nature and its ability to efficiently explore the search space make it suitable for route generation in 3D environments with obstacles and unknown structures.
- **RRT* (Rapidly Exploring Random Tree Star):** RRT* is an improvement of RRT that optimizes the trajectories generated by the original algorithm. RRT* seeks to improve the efficiency of the route found by RRT by reducing the path length and optimizing the tree structure. While RRT* can provide optimal routes, its computational complexity may significantly increase in complex 3D environments.
- **Potential Fields:** Potential fields is a path-planning technique that uses attractive and repulsive forces to guide the movement of the UAV towards the goal and away from obstacles. While potential fields can generate smooth trajectories, they may suffer from local minima and oscillations in environments with complex obstacles.

4.3.2. Selection of the RRT Algorithm

After evaluating the path-planning algorithms [41], the RRT algorithm has been selected for vehicle route generation in the context of a 3D scenario. The choice is based on the following reasons:

- RRT is a probabilistic algorithm that can efficiently explore the search space, making it suitable for unknown and complex environments.

- The random sampling nature of RRT allows for the generation of diverse trajectories and the exploration of the different regions of a space.
- RRT has a relatively simple implementation and low computational complexity compared to more advanced algorithms like RRT*.

4.3.3. Process of Route Generation and Optimization Using the RRT Algorithm

The RRT algorithm is a path-planning method extensively used in robotics and autonomous systems for navigating through high-dimensional spaces with obstacles. It is particularly valuable for its efficiency and ability to handle complex environments.

RRT is designed to efficiently explore large and high-dimensional spaces by incrementally building a tree that extends towards unexplored regions. This method is probabilistically complete, meaning that given enough time, it will find a path if one exists.

The steps of the algorithm are described below:

1. **Initialization:** The process begins with the initialization of a tree T , which initially contains only the starting position of the UAV, denoted as x_{init} .
2. **Random sampling:** In this step, the algorithm randomly samples a point x_{rand} within the search space. This random point represents a potential new position that the UAV might move to.
3. **Nearest neighbor selection:** Once a random point x_{rand} is chosen, the algorithm identifies the nearest node $x_{nearest}$ in the existing tree T . This is typically accomplished using a distance metric such as the Euclidean distance to determine the closest existing node to the new random sample.
4. **New node generation:** After identifying the nearest node, the algorithm generates a new node x_{new} . This new node is created by moving a certain step size δ from $x_{nearest}$ towards x_{rand} . This step ensures that the tree expands incrementally and steadily. The formula for the new node is as follows:

$$x_{new} = x_{nearest} + \delta \cdot \frac{x_{rand} - x_{nearest}}{\|x_{rand} - x_{nearest}\|}$$

Here, $\|x_{rand} - x_{nearest}\|$ represents the Euclidean distance between $x_{nearest}$ and x_{rand} .

5. **Collision checking:** The next crucial step involves checking for collisions. The algorithm verifies whether the path from $x_{nearest}$ to x_{new} intersects with any obstacles in the environment. If the path is collision-free, it is considered a valid extension of the tree.
6. **Tree expansion:** If no collisions are detected, x_{new} is added to the tree T , with an edge connecting it to $x_{nearest}$. This new node and edge represent a feasible segment of the UAV's path.
7. **Repeat process:** The steps of random sampling, nearest neighbor selection, new node generation, and collision checking are repeated iteratively. The process continues until the tree reaches the goal or a predefined number of iterations is completed.
8. **Path extraction:** When the tree reaches the goal, the algorithm extracts the path from the initial position x_{init} . This path is obtained by tracing back through the tree from the goal node to the start node, resulting in a series of connected nodes representing the route.

To optimize the trajectory, trajectory smoothing is performed by applying curve fitting techniques to make it smoother and eliminate sharp oscillations. Then, unnecessary nodes are removed using trajectory simplification algorithms, reducing complexity and improving flight efficiency.

In summary, the RRT algorithm seeks to generate a route that connects the initial node with multiple goal nodes in the 3D scenario, allowing the drone to complete its reconnaissance mission autonomously and avoiding collisions with the objects in the environment. Trajectory optimization ensures that the drone follows the route more

smoothly and efficiently, providing safe and effective route planning in an unknown and complex environment.

4.3.4. Parameter Settings in the RRT Algorithm

The performance of the RRT algorithm in generating effective and efficient UAV flight paths is highly dependent on the configuration of several key parameters. The following parameters were considered and adjusted to optimize the path-planning process in our study:

1. **Step size (δ):** The step size determines the distance between the nodes in the RRT tree. It represents how far the algorithm moves from the nearest node towards a randomly sampled point in each iteration.
 - **Impact:** A smaller step size results in a finer, more precise path but increases computational time and the number of nodes. Conversely, a larger step size reduces computational load but may miss narrow passages and lead to less optimal paths.
 - **Setting used:** We set the step size to 5 m, which provided a good balance between path precision and computational efficiency.
2. **Maximum number of nodes:** This parameter defines the upper limit on the number of nodes in the RRT tree.
 - **Impact:** Limiting the number of nodes helps in managing computational resources and ensuring that the algorithm terminates in a reasonable time frame. However, too few nodes might result in the incomplete exploration of the search space.
 - **Setting used:** We set the maximum number of nodes to 1000, ensuring sufficient exploration while maintaining computational feasibility.
3. **Goal bias:** The goal bias parameter defines the probability of sampling the goal point directly instead of a random point in the search space.
 - **Impact:** A higher goal bias increases the chances of quickly reaching the goal, but may reduce the algorithm's ability to explore alternative, potentially better paths. A lower goal bias promotes exploration but might slow down convergence to the goal.
 - **Setting used:** We used a goal bias of 0.1 (10%), which allowed for the balanced exploration and goal-directed growth of the tree.
4. **Collision detection threshold:** This threshold defines the minimum distance from obstacles within which a path is considered a collision.
 - **Impact:** A lower threshold increases the sensitivity to obstacles, ensuring safer paths but potentially making it harder to find feasible routes. A higher threshold reduces sensitivity, which might lead to paths that are too close to obstacles.
 - **Setting used:** We set the collision detection threshold to 3 m to ensure safe navigation around obstacles.

These parameter settings significantly influenced the performance of the RRT algorithm in our path-planning framework:

- **Path quality:** The chosen step size and goal bias facilitated the generation of smooth and direct paths, enhancing the UAV's ability to navigate complex environments efficiently.
- **Computational efficiency:** Setting a reasonable maximum number of nodes and step size ensured that the algorithm performed within acceptable time limits, making it suitable for real-time applications.
- **Safety:** The collision detection threshold ensured that the generated paths maintained a safe distance from obstacles, reducing the risk of collisions during the UAV's mission.

5. Implementation

The implementation process of the design outlined in the previous section is explained. The process is divided into two distinct parts:

- First, the implementation of image segmentation from an orthophoto and the application of a SAM model.
- Second, the generation of viewpoints around a sample structure and the subsequent generation of a route that allows for the collision-free scanning of the structure.

The tests of both the parts explained in this section are more basic tests focused on building the foundation of the system and testing that the designed solution is viable and implementable. In the Validation Section, tests with more complex figures and environments are shown, which are considered to better demonstrate the capabilities of the designed system.

5.1. Image Segmentation and Identification

The implementation of the proposed system begins with the acquisition of the orthophotos and point cloud models of the area in question. To complete the first part of the solution design, the system relies on orthophoto acquisition missions and the application of image segmentation models.

5.1.1. Orthophoto Missions

This work builds upon previous project developments, assuming that those systems function correctly and can be included without the need for new developments for their operation. To conduct initial tests and have a dataset to work with, and to test how different parameters affect the quality of the models and photographs, it is decided to start with the execution of the defined standard mission.

The mission is located in the perimeters of the Colmenarejo campus of the Carlos III University of Madrid (Figure 6). A swarm of three drones is defined to perform the necessary captures, and a rectangular area is defined that encompasses a part of the university campus, the nearby housing developments, and a large portion of undeveloped land.

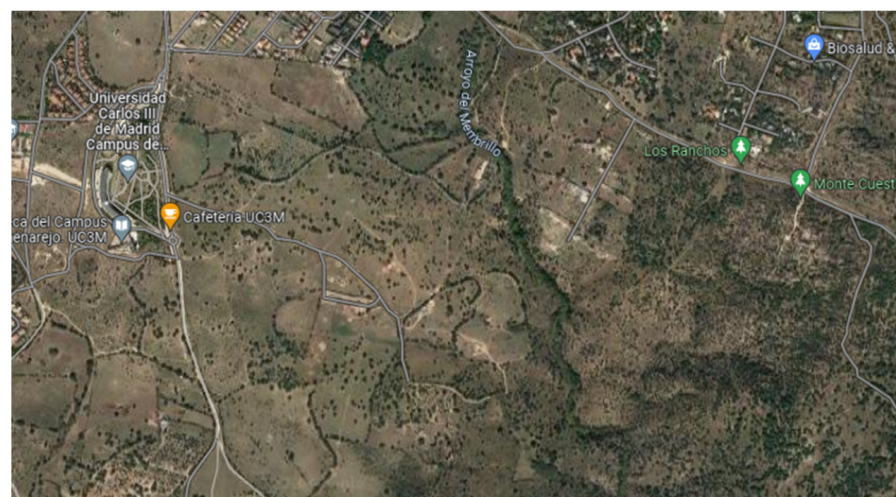


Figure 6. Aerial image of the testing ground.

Once the mission area and the height at which the drones must fly with respect to the ground (250 m) have been defined, the terrain is divided between the three drones so that each one takes photographs in different positions, and the routes they must describe and the points from which they must capture the terrain are defined. The routes followed by the drones for this specific exercise are as shown in Figure 7:

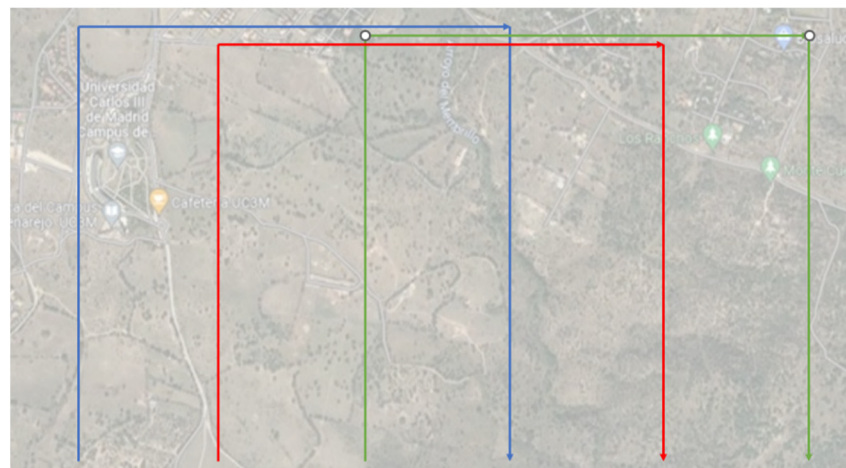


Figure 7. Drone routes on orthophotography missions.

All the missions executed with this system return the same results: the set of images obtained by each drone. As indicated in the design of the proposed solution, these images need to be processed because they do not possess the qualities of an orthophoto in this state. Additionally, the mission does not return the necessary point cloud model to define any route later on. To solve this problem, Docker machine instances with the OpenDroneMap library installed are utilized.

The OpenDroneMap system allows for the modification of numerous parameters before initiating the image analysis process to obtain different 3D models of the same terrain. In this initial test scenario, the system is configured to obtain the best-defined 3D point cloud model possible and an orthophoto of the terrain. Additionally, files in .csv and other formats were generated with information such as the geographical coordinates of the identified points in case they are needed in the segmentation and processing step of the point cloud.

The results of the two files that are of great importance in the research are shown below in Figures 8 and 9.



Figure 8. Mission result: orthophoto.

As can be observed, the results of this first test are not the most accurate when it comes to using the point cloud model to construct a route around any structure due to the lack of detail they present. Nevertheless, it is decided to continue with this sample to work on image segmentation simply to verify that the ideas presented in the previous chapter can be developed, or if it is necessary to go back and modify a part of the described system. Additionally, the point cloud file is modified and converted to .ply to be able to work with it correctly later on.



Figure 9. Mission result: point cloud.

5.1.2. Application of SAM Model

The next part of the designed system is the application of image segmentation models to allow the users who adopt this model to apply filters over large terrains and focus their reconnaissance missions on the areas of interest. When applying this segmentation model, the previously mentioned design is utilized.

The system allows users to load a map, overlay any image on it, and select the points of interest. This functionality aligns perfectly with the proposed solution, enabling users to generate a flight route around a structure depicted in an orthophoto by simply marking the object's boundaries with points on the map. The system allows loading the orthophoto obtained in the previous step and selecting any part of it through two groups of points: those indicating the pieces that are part of the structure (inbound) painted green, and those delimiting its borders and marking the rest of the image (outbound) painted red (Figure 10).

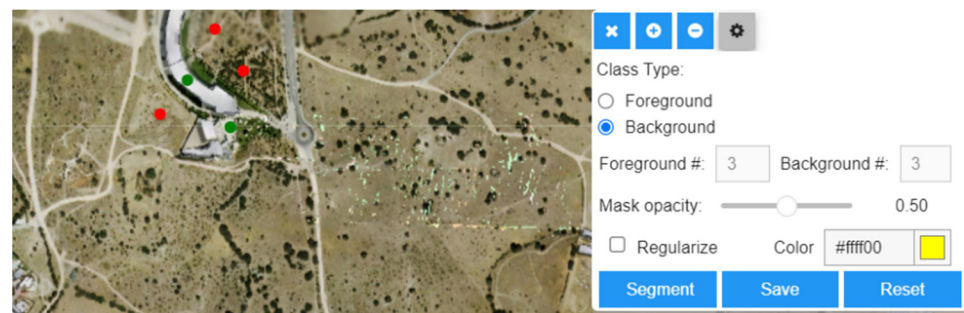


Figure 10. Example of system segmentation use.

As a result, the model returns the outline of all the structures it has recognized with their coordinates in latitude and longitude. Following the steps explained above, the point cloud is processed by applying the segmentation model's results. To select only the points that are within the contour, it is decided to select the points whose coordinates on the X and Z axes lie within the line delimited by the structure's contour, regardless of the height with respect to the Y axis at which they are located.

As an initial approximation of the system's operation, the results obtained allow the validation of whether it is possible or not to extract the point cloud that models a specific structure from the orthophoto obtained with the drone mission and the SAM segmentation models. In subsequent tests, the initial parameters will be refined with the intention of being able to handle orthophotos and point clouds in more detail to generate flight route models with greater reliability and robustness.

5.2. Generation of Viewpoints and Flight Route

This section aims to explain the implementation of the 3D data processing algorithm used for generating drone trajectories. The process followed from this point consists of the following:

- The generation of viewpoints;
- The definition of the flight route.

Below is the analysis and filtering of points in 3D meshes using clustering algorithms. The main objective is to process a 3D point cloud to identify and extract the significant clusters of proximate points. These clusters can then be used for further analysis or applications across various fields, including computer vision, image processing, and 3D graphics.

To achieve this goal, the *Open3D* library will be utilized, a powerful tool for processing and visualizing 3D data. Throughout the study, the different blocks that will compose the implementation of the algorithm will be detailed, allowing for a complete understanding of the process. These blocks can be divided into the following:

- The definition of functions;
- Data processing;
- The generation and filtering of displaced points;
- The clustering of filtered points;
- The additional filtering of centroids and second cluster.

5.2.1. Definition of Functions

In this block, two crucial functions for data processing using the *Open3D* library and the *DBSCAN* clustering technique are introduced.

The first function is *voxel_grid_downsampling*. This function is responsible for sampling points in a 3D point cloud using a voxel grid. The main goal of this process is to reduce the point cloud's density, which can be beneficial for various purposes such as improving computational efficiency and simplifying the scene's representation without losing critical information.

The second function is *min_distance_to_mesh*. This function calculates the minimum distance between a point and all the triangles forming a three-dimensional mesh. This functionality is essential for evaluating the proximity of displaced points to the mesh vertices during the filtering process.

5.2.2. Data Processing

Firstly, the initial tasks necessary to work with the 3D point cloud and prepare the data for subsequent processing are addressed. Then, each step performed in this block is explained in more detail:

1. **Reading the PLY file:** In this stage, a point cloud file in PLY format is read. The PLY (*Polygon File Format*) format is widely used to represent 3D data, such as point clouds and three-dimensional meshes. The information contained in this file represents the 3D scene to be processed. Once the file has been read, the data are stored in the variable *pcd*, representing the 3D point cloud.
2. **Downsampling of the point cloud:** Downsampling is a technique that allows reducing the number of points in a point cloud without losing too much relevant information. In this stage, downsampling is applied to the 3D point cloud using the *voxel_grid_downsampling* function. This function samples points using a voxel grid, where each voxel represents a three-dimensional cell in space. The result of downsampling is stored in the variable *pcd_downsampled*, containing the 3D point cloud with a reduced number of points.
3. **Creation of mesh using alpha shapes:** The alpha shape technique is a mathematical tool used in computational geometry to represent sets of points in three-dimensional space. In this stage, an *alpha* value is defined, a crucial parameter in mesh generation. Then, the alpha shape technique is applied to the downsampled point cloud to create

a three-dimensional mesh encapsulating and representing the points in space. The resulting mesh is stored in the variable *mesh*.

4. **Calculation of vertex normals of the mesh:** Normals are vectors perpendicular to the surface of the mesh at each of its vertices. Calculating normals is useful for determining the surface orientation and direction at each point of the mesh. In this stage, the normals of the vertices of the mesh created in the previous step are calculated, providing additional information about the scene's geometry and facilitating certain subsequent operations, such as the proper visualization of the mesh.

5.2.3. Generation and Filtering of Displaced Points

In the next step, a series of steps are performed to generate and filter displaced points from the centers of the triangles of the three-dimensional mesh. The steps taken in this block are detailed below:

1. **Creation of displaced points:** In this stage, the centers of each triangle of the previously created three-dimensional mesh are obtained. These centers represent points in space located at the center of each of the mesh's triangles. From these points, new displaced points are generated along the normals of the triangles. The normal of a triangle is a vector perpendicular to its surface and is used to determine the direction in which points should be displaced. In this case, the points are displaced 10 m along the normals, creating new points in space.
2. **Creation of displaced point cloud:** Once the displaced points have been generated, a new three-dimensional point cloud called *displaced_pcd* containing these points is created. This point cloud is a digital representation of the displaced points in space (see Figure 11).
3. **Calculation of distances:** In this stage, the distances between the displaced points and the vertices of the three-dimensional mesh are calculated. For each displaced point, the minimum distance to all the vertices of the mesh is measured. The result of these calculations is stored in the list *distances*, containing the distances corresponding to each displaced point.

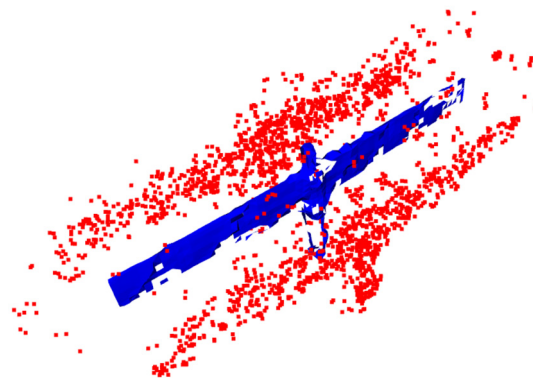


Figure 11. Display of the displaced points.

5.2.4. Clustering of Filtered Points

We perform clustering on the previously filtered points using the DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise). The process involves the following stages:

1. **Applying the DBSCAN algorithm:**

We use the DBSCAN algorithm to cluster the filtered points. DBSCAN relies on point density and is effective for finding arbitrary-shaped clusters while separating noisy points. We set the parameter *eps* to two, representing the maximum distance between points to be considered in the same neighborhood. Additionally, we set *min_samples* to one, indicating the minimum number of points required to form a cluster as one. These

parameters influence how the algorithm groups the points and affect the size and shape of the resulting clusters.

2. Separating points into clusters:

After applying DBSCAN, the filtered points are separated into different clusters based on the labels obtained from the algorithm. Each point is assigned to a cluster based on its proximity and density relative to the other points in the three-dimensional space. Close and dense points are part of the same cluster, while those further away or with low density are considered as noisy points or form individual clusters.

3. Assigning colors to clusters:

To visually distinguish the clusters, each one is assigned a random color. This color assignment facilitates the visual identification of the clusters in the dataset and provides a better understanding of the spatial distribution of the grouped points.

4. Creating point clouds for each cluster:

To visualize the clusters, a point cloud is created for each one. Each point cloud contains the points associated with the respective cluster, enabling separate visualization. This representation (Figure 12) aids in the inspection and visual analysis of the grouped data, offering insights into the spatial structure of the clusters.

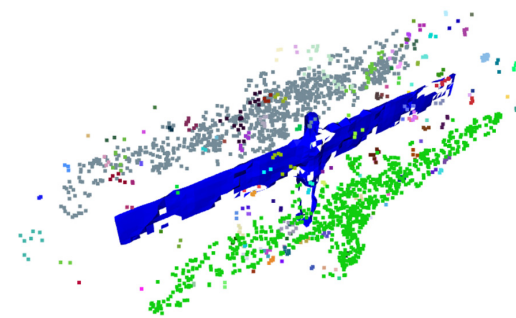


Figure 12. Initial clusters identified.

5. Calculation of cluster centroids

The centroid is calculated for each cluster, which represents the average point in the three-dimensional space of the points belonging to that cluster. The centroids are reference points that provide information about the location and general shape of each cluster (see Figure 13). These centroids are stored in a list called *centre_points*, which will be used in the later stages of the analysis and visualization process.

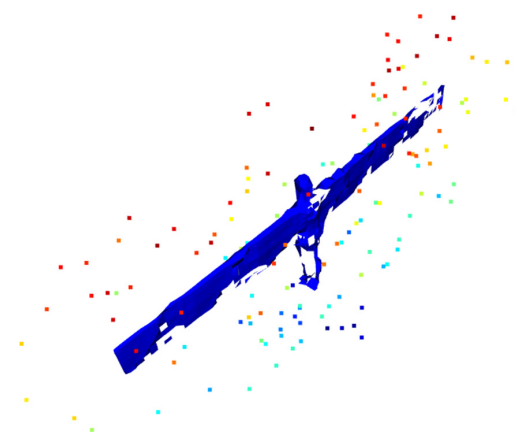


Figure 13. Central points grouping each cluster.

5.2.5. Additional Filtering of Centroids and New Clustering

We conduct an additional filtering stage on the centroids of the previously obtained clusters. The following describes each step in detail:

1. Creating a new point cloud of centroids:

A new point cloud named *center_points_pcd* is created, containing the centroids of each cluster. Centroids are representative points calculated as the average point of all the points belonging to a cluster. This point cloud of centroids provides information about the spatial location of the grouped clusters and will be used for further filtering.

2. Calculating distances between centroids and mesh vertices:

Distances between the points of the centroid point cloud and the vertices of the three-dimensional mesh are computed. This operation evaluates how close the centroids are to the surface represented by the mesh. The resulting distances are stored in a list called *distances_center_points*, containing the distances between each centroid and the mesh vertices.

3. Filtering centroids based on minimum distance to mesh:

The centroids are filtered based on the minimum required distance between them and the mesh vertices. Specifically, the centroids within less than 5 m from the mesh vertices are selected to be part of the *filtered_center_points* list. This filtering aims to remove the centroids that may be too close to the mesh surface and do not provide relevant or useful information for analysis (Figure 14).

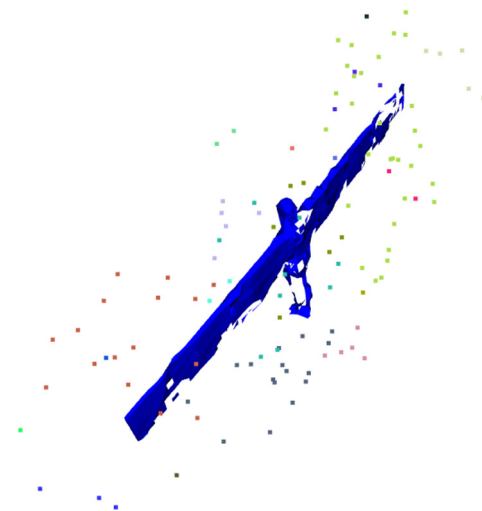


Figure 14. Filtering of central points for each cluster.

5.2.6. Second Clustering of Filtered Centroids

Finally, a second clustering stage is performed using the DBSCAN technique on the previously filtered centroids. The following outlines the steps carried out in this block:

1. Second Clustering with DBSCAN:

The DBSCAN clustering algorithm is applied again to the centroids that passed the filtering process in the previous step. This time, different parameters are used for clustering. The eps value is set to 6 m as the maximum distance between points to be considered in the same neighborhood. These values differ from those used in the first clustering, potentially resulting in clustering the centroids into different clusters.

2. Separating centroids into clusters:

After the second clustering with DBSCAN, the centroids are separated into different clusters based on the labels obtained in this process. The centroids close to each other in

space are grouped together in the same cluster, while those further apart are assigned to different clusters.

3. Assigning random colors to clusters:

Each of the clusters obtained in the second clustering is assigned a random color. This color assignment allows for the easy visualization and distinction of the different clusters during analysis.

4. Creating a new point cloud for each cluster:

From the centroids grouped into each cluster, a new point cloud is created for each one. These point clouds are stored in the list, with each element representing a cluster and containing the points belonging to that specific cluster.

5. Recalculating centroids:

After the second clustering, the centroids are recalculated for each cluster. The centroid is the average point of all the points forming a part of the cluster, representing the central or representative point of the cluster (see Figure 15).



Figure 15. Central points resulting from the second clustering process.

5.3. Flight Path Definition

This section aims to explain the process of generating a flight path for an unmanned aerial vehicle (UAV) in the context of a 3D environment. Flight path generation is a crucial stage in this project's development, as it will enable the drone to autonomously complete a reconnaissance mission while avoiding collisions with the objects in the environment.

To achieve this goal, we employ the Rapidly Exploring Tree (RRT) algorithm, widely used for path planning in complex and unknown environments.

As previously explained, the RRT algorithm is a path-planning technique based on the random sampling of the search space to construct a search tree representing the possible trajectories for the drone. The algorithm starts with an initial node representing the drone's position in the environment, and in each iteration, a random sampling point is generated in the search space. This point is then connected to the nearest node in the existing tree. If the connection line between these two points does not collide with the objects in the environment, the new node is added to the tree, and the process repeats until a certain number of nodes is reached or until a goal node is found.

5.3.1. Sampling Point Selection

In the sampling point selection phase, the RRT algorithm randomly searches for points in the 3D space of the environment to expand the search tree and generate new possible trajectories for the drone. The sampling point selection process proceeds as follows:

1. Generating sampling points:

- In each iteration of the RRT algorithm, a sampling point is randomly generated in the 3D space of the environment. This point represents a potential position to which the drone could move along its route.
 - The sampling points are generated randomly, allowing for the exploration of the different regions of the search space and the discovery of new possible trajectories.
2. **Collision checking:**
 - Once the sampling point is generated, the algorithm checks if the path between the new node and its nearest parent node collides with objects present in the environment. Collision detection functions are used to evaluate the distance between the new node and the objects represented in the 3D mesh of the environment.
 - If the path between the new node and its parent node does not collide with objects, it means that the drone can fly from its current position to the new sampling point without encountering obstacles along the way.
 3. **Distance constraint compliance:**
 - In addition to collision checking, the algorithm also evaluates whether the new node complies with distance constraints. That is, it checks if the distance between the new node and its nearest parent node is less than or equal to a predetermined distance value.
 - Distance constraint is essential to ensure that the drone makes reasonable movements and does not make excessively large jumps in space.
 4. **Addition to the RRT Tree:**
 - If the new node complies with the distance constraints and does not collide with objects, it is added to the RRT tree as a new node. The nearest parent node to the new node becomes its predecessor in the tree, allowing for tracking the path from the initial node to the new node.
 - As new nodes are generated and added, the RRT tree expands and explores the different possible trajectories for the drone.

The process of sampling point selection, along with collision checking and distance constraint compliance, is crucial for constructing a meaningful RRT tree and finding viable and safe trajectories for the drone. By selecting sampling points and performing collision checks, the algorithm can efficiently explore the search space and find routes that enable the drone to complete its reconnaissance mission without collisions and autonomously.

5.3.2. Path Search and Optimization

After building the RRT tree and reaching a goal node representing the desired destination point, an initial trajectory is created connecting the initial node with the goal node. However, this initial trajectory may contain redundant nodes and may not represent the optimal route for the UAV. Therefore, it is necessary to optimize the obtained trajectory using smoothing and unnecessary node removal techniques. This process is carried out in two stages:

1. **Path smoothing:**
 - In this stage, a smoothing technique is applied to the initial trajectory generated by the RRT tree. The goal of smoothing is to make the trajectory smoother and eliminate possible sharp oscillations or abrupt changes in direction. This not only improves the aesthetics of the trajectory but can also have a positive impact on flight efficiency and reduce energy consumption.
 - A commonly used technique for trajectory smoothing is fitting a curve through the points of the initial trajectory. This can be achieved through interpolation or curve approximation techniques. By fitting a curve to the trajectory points, a new smoothed trajectory is obtained that follows the general direction of the original trajectory but with smoother curves.

- In this case, trajectory smoothing is performed using a spline fitting algorithm. A spline represents a mathematical curve that passes through a set of given points and creates a smooth and continuous path between them. By obtaining a smoother trajectory, the drone can follow the route more smoothly and steadily, resulting in a more efficient and safe flight.
- The spline fitting process involves the following steps:
 - Taking the points of the initial route generated by the RRT tree.
 - Applying the spline fitting algorithm, which calculates a curve that fits the points optimally, attempting to minimize deviations between the curve and the points.
 - The resulting curve becomes the new smoothed trajectory that the drone will follow during flight.

2. Unnecessary node removal:

- Once the trajectory has been smoothed, it may still contain unnecessary nodes that do not provide relevant information for the drone's flight. These redundant nodes may arise due to the hierarchical structure of the RRT tree and how the initial trajectory is constructed.
- To optimize the trajectory, the unnecessary nodes are removed. This is achieved by applying trajectory simplification algorithms, such as the Douglas–Peucker algorithm, which reduces the number of points in the trajectory while maintaining its general shape. Removing the unnecessary nodes helps reduce the complexity of the trajectory and improves system efficiency by simplifying the calculations required to follow the route.
- The process involves the following steps:
 - Analyzing the initial trajectory generated by the RRT tree.
 - Identifying those nodes that do not provide relevant information for reaching the desired destination point.
 - These unnecessary nodes are removed from the trajectory, resulting in a simplified and more direct trajectory.

By combining trajectory smoothing and the removal of unnecessary nodes, an optimized path is obtained that represents the best approximation of the optimal route for the drone to complete its reconnaissance mission. The optimized trajectory is easier to follow and provides a more efficient and safer route around the obstacles and structures in the environment (Figure 16).

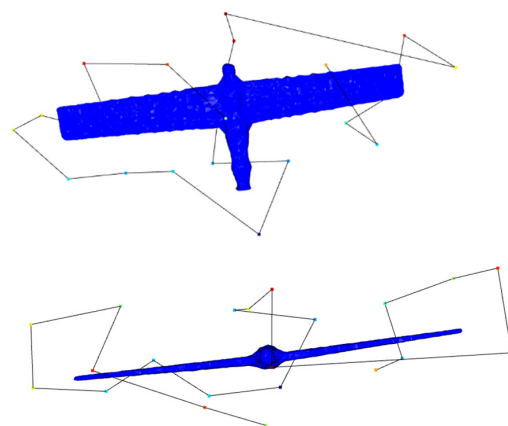


Figure 16. Route generated. Aerial and front view.

The following qualitative results are obtained:

- The framework successfully identified multiple viewpoints around the statue, ensuring comprehensive coverage.

- The flight path generated allowed the UAV to navigate around the statue smoothly, with no detected collisions.
- The captured images had significant overlap, which is crucial for high-quality 3D reconstruction.
- The 3D model generated from these images was detailed and accurate, capturing the intricate features of the statue.

6. Validation

In the following section, a new use case is presented to verify that the designed system functions effectively in different scenarios and ensures that the results are consistent and align with the real models of the structures.

The methodology followed in this use case is the same as that outlined in the Implementation Section. Therefore, this section will focus on presenting the results obtained in the new scenario and discussing any particularities it may present.

In this use case, the objective is to verify that the part of the system responsible for generating viewpoints and collision-free flight paths adapts to different object structures and produces good results for objects other than the angel figure shown in the previous section. For this case, the building shown in Figure 17 is chosen.

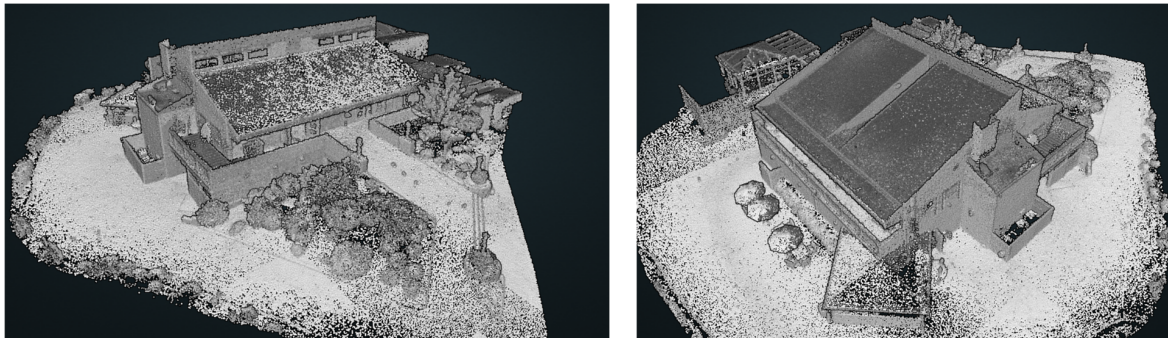


Figure 17. Point cloud.

The process of generating and filtering displaced points is conducted to obtain a relevant point cloud that will serve to generate viewpoints and plan the drone's flight path for the reconnaissance mission.

Finally, the filtered displaced points are visualized along with the three-dimensional mesh using the Open3D library. The displaced points are shown in red, allowing for the clear and prominent visualization of the filtered points in relation to the mesh.

Subsequently, point clouds are created for each cluster, grouping the points associated with each one. This enables them to be visualized separately and facilitates the analysis of the spatial structure of the clusters (Figure 18).

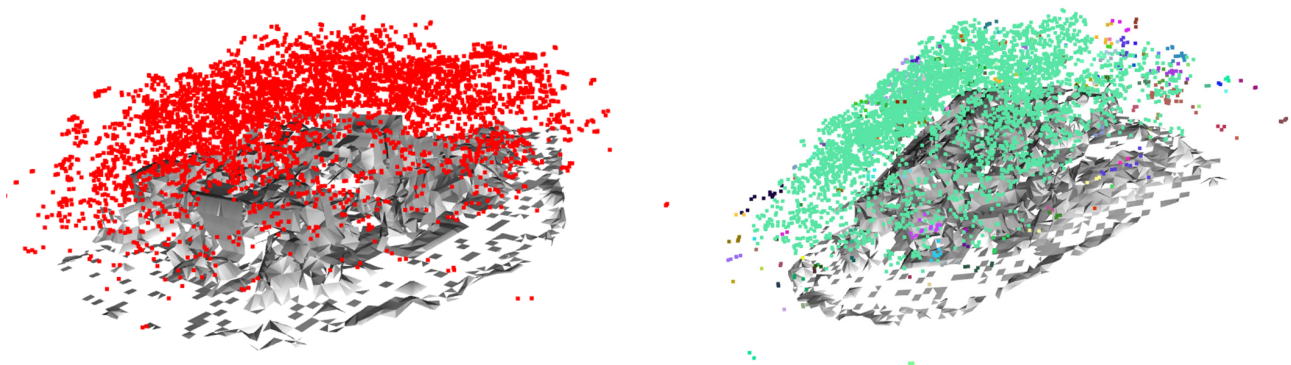


Figure 18. Display of the displaced points (left). The initial clusters identified (right).

In the same manner as previously presented, the centroid for each cluster is calculated, representing the average point in the three-dimensional space of the points belonging to that cluster. The centroids provide information about the location and general shape of each cluster. An additional filtering is performed on the obtained centroids, followed by a second clustering using DBSCAN with different parameters to obtain different groupings. Next, the DBSCAN algorithm is applied again to the filtered centroids, using different “eps” values. This allows the centroids to be grouped into different clusters (Figure 19).

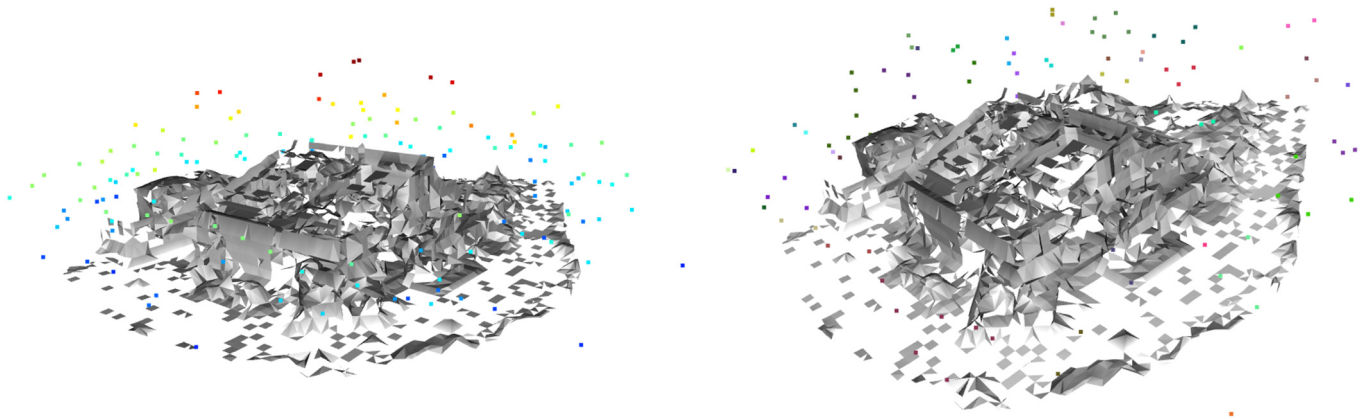


Figure 19. Central points grouping each cluster (left). The filtering of the central points of each cluster (right).

A random color is assigned to each cluster resulting from the second clustering, and a new point cloud is created for each cluster, facilitating their visualization and analysis. After the second clustering, the centroid for each cluster is recalculated, providing a representative point for each grouping (see Figure 20).

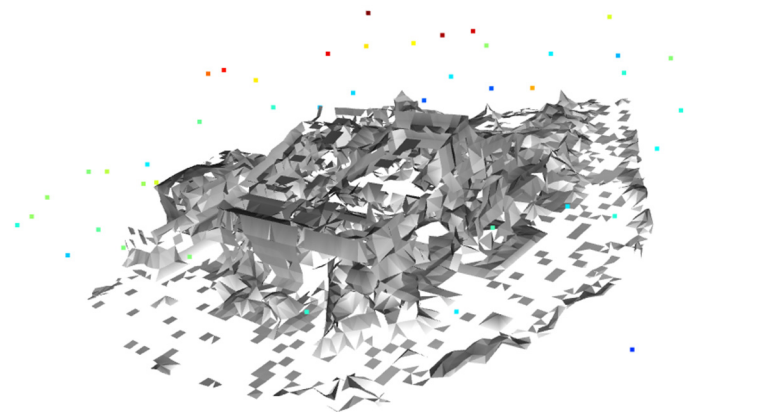


Figure 20. Focal points resulting from the second clustering process.

The final step involves defining the flight path. This will follow the same structure presented in the Implementation Section. Therefore, only the main stages necessary for defining this route will be included below.

In the Flight Path Definition, the RRT algorithm is used to generate a path for a drone in a 3D scenario. The goal is for the drone to complete a reconnaissance mission autonomously while avoiding collisions with the objects in the environment.

The route generation process begins with the random selection of the sampling points in the 3D space of the scenario. These points represent the potential positions to which the drone could move. It is checked whether the path between the new point and its nearest parent node collides with objects and whether it meets the distance constraints to ensure reasonable movements.

Once new nodes are generated and added, the RRT tree expands and explores different possible trajectories for the drone.

After constructing the RRT tree and reaching the target node, an initial trajectory is created. The trajectory is then optimized by smoothing using a spline fitting algorithm, improving the aesthetics, efficiency, and stability of the flight. Additionally, the unnecessary nodes are removed to reduce complexity and obtain a more direct route.

In conclusion, the Flight Path Definition using the RRT algorithm allows for obtaining an optimized and safe route for the drone to carry out its reconnaissance mission autonomously in complex environments, avoiding obstacles and following a smooth and efficient trajectory (see Figure 21).

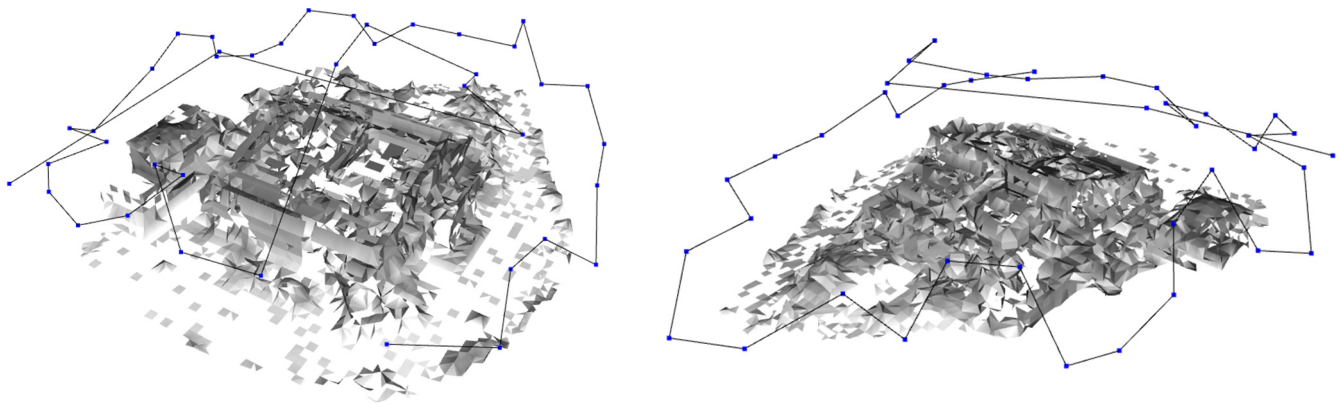


Figure 21. Views of the generated route.

The following qualitative results are obtained:

- The framework effectively adapted to the complex architecture of the building, generating appropriate viewpoints.
- The UAV followed the planned path without any collisions, successfully navigating around the building's features.
- The images captured provided a comprehensive view of the building from various angles, essential for 3D modeling.
- The resulting 3D reconstruction was accurate, capturing the building's structural details comprehensively.

7. Future Research Directions

Although this study presents a robust framework for generating collision-free flight paths for UAVs in 3D reconstruction missions, several limitations and opportunities for future research have been identified.

One key limitation is the dependency on high-quality orthophotos and point clouds. Variations in the data quality can significantly impact the accuracy of the generated flight paths and 3D models. Future research should focus on improving data pre-processing techniques and exploring ways to enhance the robustness of the system against varying data quality. Additionally, incorporating advanced data cleaning and normalization processes can help mitigate the effects of inconsistent input data.

Another area for further exploration is the integration of additional sensor types. While this study primarily utilized RGB cameras and LiDAR, incorporating thermal cameras, multispectral sensors, and advanced imaging technologies could provide richer datasets and improve the fidelity of the 3D reconstructions. The use of these diverse sensors would allow for a more comprehensive understanding of the environment, capturing different aspects such as temperature variations and material properties.

The computational complexity of the path-planning algorithms also presents a limitation. Although the RRT algorithm was selected for its efficiency in complex environments,

optimizing these algorithms to reduce computational overhead while maintaining accuracy is a critical area for future research. Techniques such as parallel processing, heuristic optimizations, and machine learning-based enhancements could be explored to improve algorithm performance.

Moreover, real-world testing and validation are essential to ensure the system's effectiveness in diverse operational scenarios. Field tests with actual UAV deployments would provide valuable insights into the practical challenges and help refine the system further. These tests should include varied environments, from urban landscapes to remote natural settings, to evaluate the robustness and adaptability of the framework.

In addition to these advancements, several specific developments and testing protocols are planned for swarm deployment. The development and integration of advanced swarm coordination algorithms, such as leader–follower dynamics and collaborative task allocation, will ensure efficient communication and coordination among multiple UAVs. These algorithms will optimize task distribution and enhance the overall efficiency of the swarm during reconnaissance missions. Furthermore, implementing real-time dynamic obstacle detection and avoidance techniques will leverage sensor fusion from multiple UAVs, combining data from LiDAR, cameras, and other onboard sensors to create a comprehensive understanding of the environment and enable adaptive path planning. Throughout the testing phases, data on mission completion time, energy consumption, collision rates, and data capture accuracy will be collected and analyzed. These metrics will provide insights into the efficiency and effectiveness of the swarm deployment strategies.

By addressing these limitations and exploring these research directions, the capabilities of UAVs in autonomous 3D reconstruction can be significantly enhanced, paving the way for more advanced and reliable applications in various fields.

8. Conclusions

This research has demonstrated the feasibility and effectiveness of utilizing autonomous drones for the 3D reconstruction of buildings and other structures. The developed system successfully integrates various advanced technologies, including image segmentation, point cloud processing, and route optimization algorithms, to achieve detailed and accurate 3D models. The key findings and contributions of this study are outlined below.

The integration of orthophotos and point clouds proved to be a robust method for capturing and processing 3D data. By leveraging these techniques, the system can generate high-resolution orthophotos and detailed point clouds, which are essential for accurate 3D reconstruction. The use of Docker and the *OpenDroneMap* library facilitated the efficient processing and conversion of aerial images into usable formats, further enhancing the system's capabilities.

The application of the Segment Anything Model (SAM) for image segmentation significantly improved the accuracy and efficiency of identifying structures within the orthophotos. This allowed for the precise delineation of the areas of interest, ensuring that the drone's flight path could be optimized to avoid collisions while capturing all the necessary details. The ability to segment images and maintain geolocation information enabled the seamless integration of point cloud data, enhancing the overall reconstruction process.

The methodology for generating viewpoints and flight routes was rigorously tested and validated. By employing clustering algorithms and the RRT path-planning technique, the system could generate collision-free flight paths that optimized the drone's trajectory around structures. This approach ensured the comprehensive coverage of the target areas, capturing the multiple perspectives necessary for high-fidelity 3D models.

The research highlighted the importance of balancing detail and efficiency in drone operations. By calculating the optimal distances for capturing images and utilizing geometric principles to maximize the coverage, the system could efficiently gather data without compromising safety or quality. This balance is crucial for practical applications in various fields, from construction and urban planning to archaeological documentation and environmental monitoring.

In summary, this study has established a solid foundation for using autonomous drones in 3D reconstruction tasks. The developed system showcases significant advancements in autonomous flight path planning, image segmentation, and data processing, contributing valuable insights and tools for future research and practical applications. The continuous evolution of these technologies promises to further enhance the precision, efficiency, and safety of 3D reconstruction efforts across diverse sectors.

Author Contributions: Conceptualization, C.G.A.; methodology, R.D.-A.J.; software, F.P.M.; validation, M.Z.S.; formal analysis, C.G.A.; investigation, C.G.A.; resources, R.D.-A.J.; data curation, F.P.M.; writing—original draft preparation, C.G.A.; writing—review and editing, M.Z.S.; visualization, F.P.M.; supervision, R.D.-A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article. Any further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [CrossRef]
- Erdelj, M.; Natalizio, E.; Chowdhury, K.R.; Akyildiz, I.F. Help from the Sky: Leveraging UAVs for Disaster Management. *IEEE Pervasive Comput.* **2017**, *16*, 24–32. [CrossRef]
- DPedroche, S.; Amigo, D.; García, J.; Molina, J.M. Architecture for Trajectory-Based Fishing Ship Classification with AIS Data. *Sensors* **2020**, *20*, 3782. [CrossRef] [PubMed]
- Blanco, M.T. *Generación y Explotación de Ortofotografías Mediante Enjambre de Drones Simulados*; Universidad Carlos III de Madrid: Madrid, Spain, 2022.
- Gómez-Calderrada, J.L. *Reconstrucción y Análisis de Objetos 3D con UAV*; Universidad Carlos III de Madrid: Madrid, Spain, 2022.
- Amigo, D.; García, J.; Molina, J.M.; Lizcano, J. UAV Simulation for Object Detection and 3D Reconstruction Fusing 2D LiDAR and Camera. In Proceedings of the 17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022), Salamanca, Spain, 5–7 September 2022.
- Yang, L.; Qi, J.; Xiao, J.; Yong, X. A literature review of UAV 3D path planning. In Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014.
- Aharchi, M.; Kbir, M.A. A Review on 3D Reconstruction Techniques from 2D Images. In *Innovations in Smart Cities Applications Edition 3—The Proceedings of the 4th International Conference on Smart City Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020.
- Santana, L.V.; Brandão, A.S.; Sarcinelli-Filho, M. Outdoor waypoint navigation with the AR.Drone quadrotor. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.
- Fuqiang, L.; Runxue, J.; Hualing, B.; Zhiyuan, G. Order Distribution and Routing Optimization for Takeout Delivery under Drone–Rider Joint Delivery Mode. *J. Theor. Appl. Electron. Commer. Res.* **2024**, *19*, 774–796. [CrossRef]
- Fuqiang, L.; Weidong, C.; Wenjing, F.; Hualing, B. 4PL Routing Problem Using Hybrid Beetle Swarm Optimization. *Soft Comput.* **2023**, *27*, 17011–17024.
- Tongren, Y.; Fuqiang, L.; Suxin, W.; Leizhen, W.; Hualing, B. A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage. *J. Ind. Manag. Optim. (JIMO)* **2023**, *19*, p1.
- Kaur, D.; Kaur, Y. Various Image Segmentation. *Int. J. Comput. Sci. Mob. Comput.* **2014**, *3*, 809–814.
- Zaitoun, N.M.; Aqel, M.J. Survey on Image Segmentation Techniques. *Procedia Comput. Sci.* **2015**, *65*, 797–806. [CrossRef]
- Meta AI—Introducing Segment Anything: Working toward the First Foundation Model for Image Segmentation. Available online: <https://segment-anything.com/> (accessed on 28 July 2023).
- Hohle, J. Experiences with the Production of Digital Orthophotos. Available online: https://www.asprs.org/wp-content/uploads/pers/1996journal/oct/1996_oct_1189-1194.pdf (accessed on 5 April 2023).
- Agüera-Vega, F.; Carvajal-Ramírez, F.; Martínez-Carricondo, P. Accuracy of Digital Surface Models and Orthophotos Derived from Unmanned Aerial Vehicle Photogrammetry. *J. Surv. Eng.* **2017**, *143*, 04016025. [CrossRef]
- Janowski, A.; Nagrodzka-Godycka, K.; Szulwic, J.; Ziółkowski, P. Remote sensing and photogrammetry techniques in diagnostics. *Comput. Concr.* **2016**, *18*, 405–420. [CrossRef]
- Dering, G.M.; Micklethwaite, S.; Thiele, S.T.; Vollgger, S.A.; Cruden, A.R. Review of drones, photogrammetry and emerging sensor technology for the study of dykes: Best practises and future potential. *J. Volcanol. Geotherm. Res.* **2019**, *373*, 148–166. [CrossRef]

20. Weiss, U.; Biber, P. Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robot. Auton. Syst.* **2011**, *59*, 265–273. [\[CrossRef\]](#)
21. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [\[CrossRef\]](#)
22. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [\[CrossRef\]](#)
23. Sudderth, E.B. Graphical Models for Visual Object Recognition and Tracking. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.
24. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment Anything. Cornell University. In Proceedings of the Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
25. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaria, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Abdullahi, H.S.; Sheriff, R.E.; Mahieddine, F. *Convolution Neural Network in Precision Agriculture for Plant Image Recognition and Classification*; University of Bradford, School of Engineering and Computer Science: Bradford, UK, 2017.
27. Lee, D.T.; Schachter, B.J. Two algorithms for constructing a Delaunay triangulation. *Int. J. Comput. Inf. Sci.* **1980**, *9*, 219–242. [\[CrossRef\]](#)
28. Newman, T.S.; Yi, H. A survey of the marching cubes algorithm. *Comput. Graph.* **2006**, *30*, 854–879. [\[CrossRef\]](#)
29. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Eurographics Symposium on Geometry Processing, Cagliari, Italy, 26–28 June 2006.
30. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Samet, H. An Overview of Quadrees, Octrees, and Related Hierarchical Data Structures. In Proceedings of the Theoretical Foundations of Computer Graphics and CAD, Ciocco, Italy, 4–17 July 1988.
32. Adamopoulos, E.; Rinaudo, F. UAS-Based Archaeological Remote Sensing: Review, Meta-Analysis and State-of-the-Art. *Drones* **2020**, *4*, 46. [\[CrossRef\]](#)
33. Hong, B.; Ngo, E.; Juarez, J.; Cano, T.; Dhoopar, P. *Simultaneous Localization and Mapping for Autonomous Navigation of UAVs in GPS-Denied Indoor Environments*; California State Polytechnic University: Pomona, CA, USA, 2021.
34. Zualkernan, I.; Abuhani, D.A.; Hussain, M.H.; Khan, J.; ElMohandes, M. Machine Learning for Precision Agriculture Using Imagery from Unmanned Aerial Vehicles (UAVs): A Survey. *Drones* **2023**, *7*, 382. [\[CrossRef\]](#)
35. Buyukdemircioglu, M.; Kocaman, S. Deep Learning for 3D Building Reconstruction: A Review. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *43*, 359–366. [\[CrossRef\]](#)
36. Maboudi, M.; Homaei, M.; Song, S.; Malihi, S.; Saadatseresht, M.; Gerke, M. A Review on Viewpoints and Path-planning for UAV-based 3D Reconstruction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *16*, 5026–5048. [\[CrossRef\]](#)
37. Yaacoub, J.-P.; Noura, H.; Salman, O.; Chehab, A. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet Things* **2020**, *11*, 100218. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Bircher, A.; Alexis, K.; Burri, M.; Oettershagen, P.; Omari, S.; Mantel, T.; Siegwart, R. Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
39. Song, C.; Chen, Z.; Wang, K.; Luo, H.; Cheng, J.C. BIM-supported scan and flight planning for fully autonomous LiDAR-carrying UAVs. *Autom. Constr.* **2022**, *142*, 104533. [\[CrossRef\]](#)
40. Shang, Z.; Shen, Z. Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures. *Complex Intell. Syst.* **2022**, *9*, 909–926. [\[CrossRef\]](#)
41. Zammit, C.; van Kampen, E.-J. Comparison between A* and RRT Algorithms for UAV Path Planning. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.