ROBOTICS AND MECHATRONICS

Texture Based Autonomous Reconstruction of 3D Point Cloud Models using Fully Actuated UAVs

R. (Rajavarman) Mathivanan

MSC ASSIGNMENT

Committee:
A. Franchi, PhD
ir. R.A.M. Rashad Hashem
dr. D. Bicego
dr. ir. D.C. Mocanu

July 2020

021RaM2020
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

Three-dimensional(3D) point cloud model of an object is essential in various applications, such as surface inspection, physical interaction with the target of interest, obstacle detection for navigation, etc. Recent progress in Simultaneous Localization And Mapping (SLAM) algorithms makes it possible to perform point cloud model reconstruction in real-time. Integrated with Unmanned Aerial Vehicles (UAVs), point cloud model reconstruction research has witnessed new possibilities. This master thesis proposes and validates an approach for ariel robotic system to autonomously build a 3D point cloud model of a 3D object using unmanned aerial vehicle (UAV), without having prior information about the object. The proposed method relies on the textures on the surface of the object for UAV coverage path planning. The reconstruction process comprises of two flight modes, initial arbitrary flight and exploration flight.The 3D point cloud generated from initial arbitrary flight undergoes spatial segmentation to determine the uncovered regions to explore and improve the point cloud model better. The identified low dense regions during the initial flight are converted into the waypoints for the UAV to navigate. These waypoints are positioned in such a way that the trajectories are obstacle free. Polar coordinate system used around the object is used to achieve the obstacle free path. There is trivial solution where a helical trajectory around an object of interest results in a perfect 3D point cloud model. However, there are some drawbacks: 1) the measurements of the object should be known before the start of trajectory to reach the maximum height and to cover the circumference of the object without colliding with the object; and 2) in a single flight, either the overlap of the field of view on the surface of the object could be avoided or the coverage of uncovered regions out of the helical path shall be satisfied. Nevertheless, the point cloud model obtained from helical trajectory experiment with sufficient number of rotations is set as a benchmark solution to evaluate our proposed method. The reconstruction completeness is evaluated using the Iterative Closest Point (ICP) algorithm which compares the point cloud model obtained using the bench mark solution with the proposed method. The proposed algorithm was simulated within the ROS system using realistic parameters of fully-actuated hexarotor UAV and a stereo vision camera for 3D perception. The simulation results showed that the proposed method, unlike trivial helical trajectory method, completes the point cloud model construction process autonomously without a priori knowledge about the object of interest. In addition to that, the proposed method completes the point cloud reconstruction in less time and high quality than the aforementioned helical solution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Modelling a three-dimensional (3D) object or a structure is an essential area of research in the field of computer vision and robotics. It is a process of representing a surface as a point cloud model. Physical interaction with the surface by a manipulator, investigation of surface quality, obstacle detection during autonomous navigation and creating a realistic city map for game engines are some of the notable applications of 3D modelling. Computer-Aided Design (CAD) as in [5] is one way of creating a 3D model. It requires a highly skilled modeller for execution. An alternate method is surface reconstruction through sensor perception. Commonly used sensors are Light Detection And Ranging(LiDAR), Radio Detection And Ranging(RADAR) and stereo cameras. A 3D object scanners are sufficient to obtain the point cloud model of small objects without much effort. However, for enormous man-made structures like skyscrapers, windmills and rocket launch pads, scanners are not large enough to solve the purpose and neither a ground based vehicles can reconstruct the structures which are tall; instead, they could be replaced with a fully actuated unmanned aerial vehicle (UAV) attached with sensors. A fully actuated UAV allows for a variety of maneuvers that includes tilted flight, inclined hovering and eliminate the use of gimbals for sensors which is not possible with under actuated airborne. Massive man-made structures require a high level of structural investigation during its construction and maintenance. It is hardly possible for humans to reach out to high altitude for inspection without expensive infrastructural support.This is possible only if an arieal medium approach is used to accomplish the task. Surface reconstruction shall be performed either using manually controlled UAVs or autonomously. An autonomous UAVs may accomplish this task more effectively than a manually controlled UAVs because it does not demand a trained operator to supervise reconstruction. It is a confined process between take-off and landing. An appropriate application to support the aim of this thesis work is surface modelling of 3D printed dome-shaped structures, as shown in Figure 1.1. National Aeronautics ansd Space Association (NASA) held a contest to design home on Mars [1]. These structures are expected to be durable and robust enough to withstand the alien weather conditions. A minor error in 3D printing

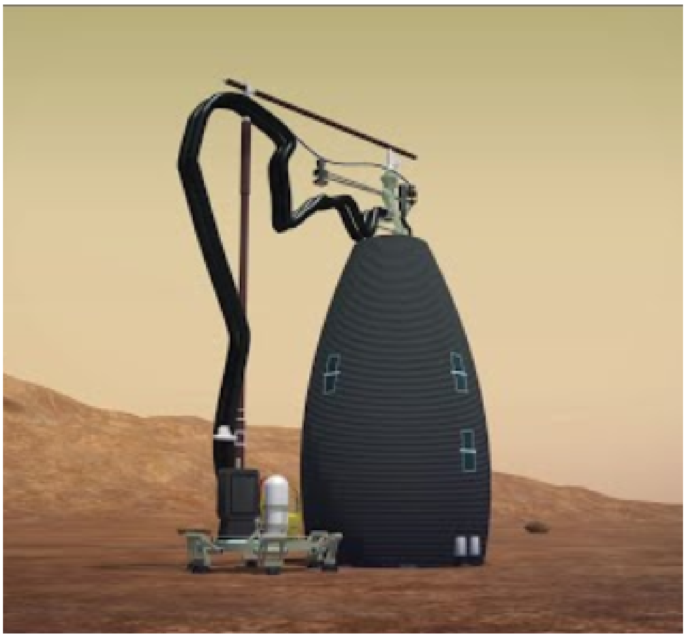


Figure 1.1: An animated model of 3D printed structure on Mars [1].

would devastate the life of astronauts. So a careful inspection of the surface is crucial in a project that demands high safety. Involving humans to inspect these tall structures in Mars is a life-risking task. This application illustrates the importance of using UAVs for surface reconstruction. Figure 1.2 shows another area of application where three workers are cleaning the glass surface of a skyscraper. Workers shall be replaced by an UAV with a manipulator that physically interacts with the surface for cleaning. The surface information is obtained through 3D point cloud reconstruction process.

There are two ways to perform autonomous surface reconstruction, offline or model-based and online or non-model based reconstruction. The model-based method demands an exact model of the object of interest and map of the surrounding where the UAV operates. Whereas, the online reconstruction process relies on sensors' data that perceive surrounding information. Thus, exploring an unknown structure in an unknown environment autonomously, without prior information (model) is more challenging than the offline method. Autonomous surface reconstruction process involves multiple iterations of the following sub-processes: 1) Generating point cloud from the surface features (perception); 2) Determining the current position and orientation of UAV with respect to the object (localization); 3) Planning the coverage path with an appropriate set of way-points for the UAV (planning), and performing a smooth trajectory to different way-points (Navigation). The explanation of these sub-processes is as follows.

Figure 1.2: Three workers cleaning the glass surface of a skyscraper. (Image from Shutterstock)

1.1.1 Perception

Point cloud acquisition system requires a surface that reflects light or beam of rays and a sensor that receives the light for processing. There are four types of techniques for point cloud acquisition. 1) Image-based method; 2) Light Detection And Ranging system (LiDAR); 3) synthetic Aperture Radar (SAR) system; 4) Camera-based method. Image-based methods as described in [6] generates a point cloud from stereotype images acquired through the camera using photogrammetry concepts. The point cloud shall be generated after the image acquisition process. This method is called the passive approach because the image acquisition and reconstruction process are not executed simultaneously. SAR system uses RADAR for point cloud reconstruction. As explained in [7] the radar moves across the object-of-interest while transmitting a successive pulse of radio waves. The pulses are received back to obtain the physical information of the surface. This method is generally used for remote sensing and reconstruction of the landscape. LiDAR-based systems as in [8] beam laser light source on the object surface and sense the reflected beam to generate point-cloud. Camera-based systems as used in [9] sample the textures on the surface of the object to generate point-cloud. Unlike image-based method, the point clouds are simultaneously generated using the image output from the stereo camera. This method is called the active approach. There are two types of camera-based sensors, monocular and stereo sensors. Of these two, the stereotype sensor produces three-dimensional points which includes the depth information of a point. The depth information of a point is essential for navigation, obstacle avoidance, physical interaction and many more applications. In general, among all the methods mentioned above, stereo camera sensors are economical, less computational and suitable for small-sized UAVs.

1.1.2 Localization and Mapping

An autonomous non-model based reconstruction using a UAV requires awareness about the current location of the UAV with respect to the point cloud map it generates. This phenomenon is called localization and Mapping, which is performed by (Simultaneous Localization And Mapping) SLAM algorithms. Several open-source SLAM algorithms performs point cloud extraction using camera or LiDAR input data and then execute localization and mapping process using the extracted point cloud. SLAM algorithm is also made available for monocular and stereo vision camera. Working of the SLAM algorithm is explained in Section 2.2.

1.1.3 Planning and Navigation

Coverage path planning(CPP) is a process of deriving several viewpoints for the UAV in such a way that, on reaching out to these viewpoints, the UAV scans the complete surface area of the object or the surrounding environment, avoiding obstacles on its way. As discussed earlier, the categorization of model-based and non-model based methods comes under this topic coverage path planning. The model-based method is otherwise called the offline method, and the non-model-based method is called online method. In the offline method, a model of the static object or map of an environment (including the position of an obstacle) is required as an input to perform coverage path planning, which ultimately leads to a global-optimal solution. Whereas in the online method, the real-time sensor measurements are used to perform path planning during the flight. In this method, knowledge about the environment is uncertain. This uncertainty leads to a local-optimal solution at its best. However, online path planning method has several advantages over the offline path planning method. For instance, in an exploration task, where the exact map of the surrounding or the accurate model of the object of interest is unavailable, an online coverage path planning method is the best choice. It is flexible in the dynamic environments, acquiring a reference model involves a high level of human effort, which is obsolete in the online-based method. In the online-based method, simultaneous localization and mapping processes are inevitable, which shall be supported by several SLAM algorithms. Figure 1.3 and Figure 1.4 shows the block diagram of online and offline path planning method respectively.

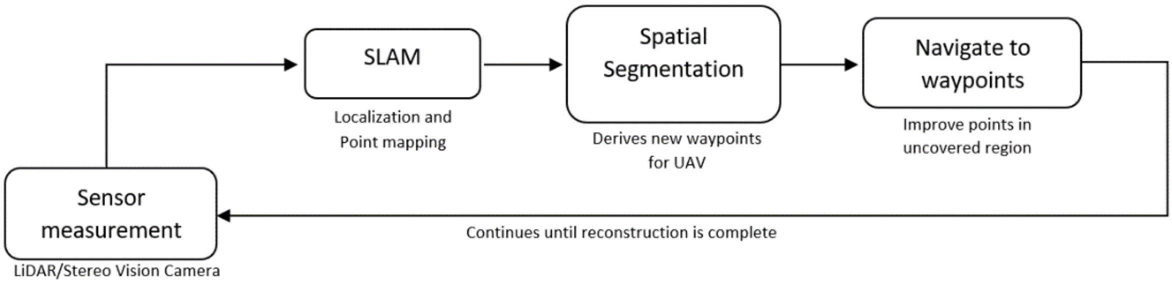


Figure 1.3: A block diagram depicting online path planning method for surface reconstruction.

Figure 1.4: A block diagram depicting offline path planning method for surface reconstruction.

Apart from the above-discussed advantages, there are challenges in the online path planning method. It is uncertain to ensure that the aim of the reconstruction process will be accomplished yet having a solution. It does not have access to a global perspective of the environment or an object of interest. Instead, it can only run the decision-making process on the go with a partial view of the subject in addition to obstacle detection. The computation cost increases at every step, as the decision-making process has to consider the updated point cloud input and obstacle information. The literature study on various online path planning algorithms is discussed in the following section.The afore-mentioned classification of 3D reconstruction methods are classified as shown in Figure 1.5. The flow diagram explains different kinds of reconstruction methodologies that are possible. The green shade in the flow chart implies the classification specific to this thesis work.

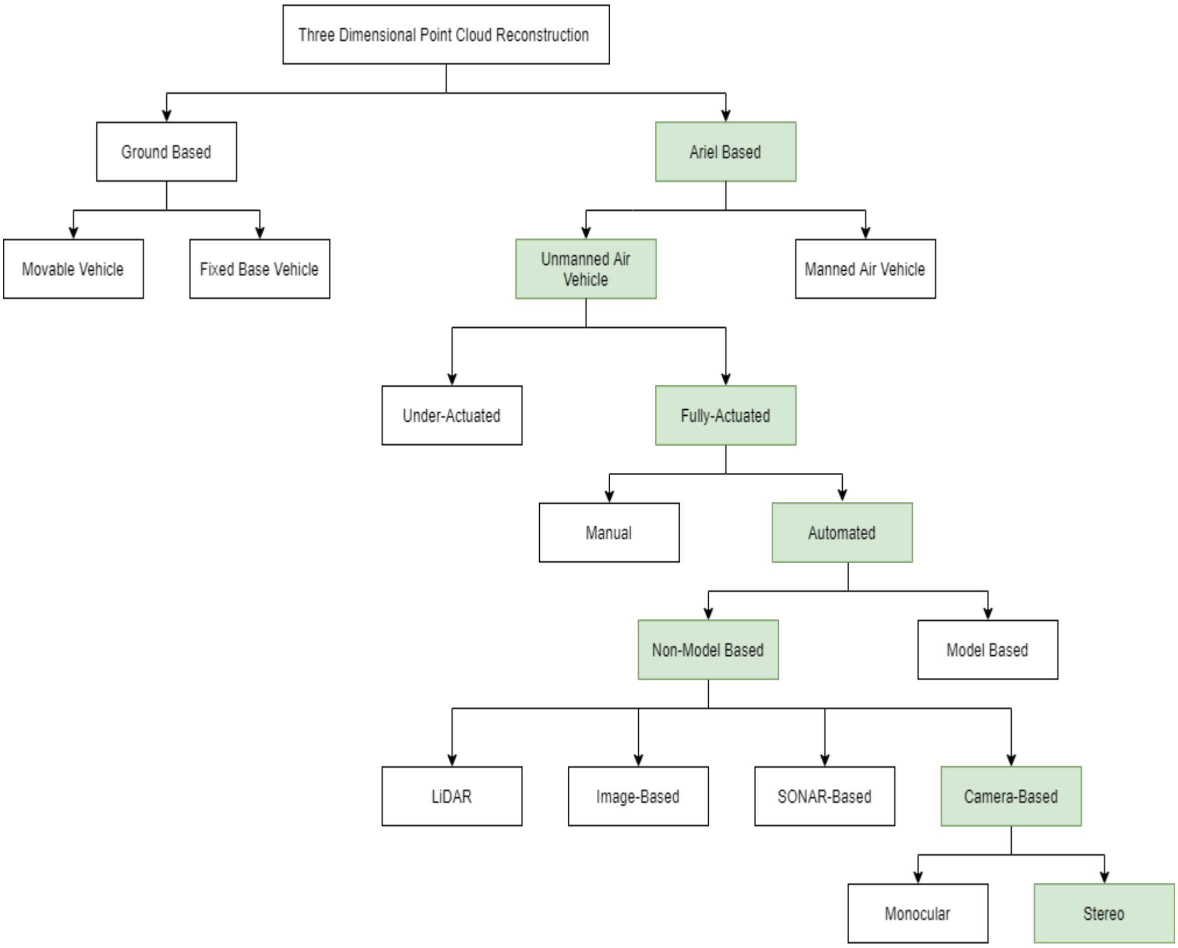


Figure 1.5: Classification of 3D reconstruction process.Green shade denotes the thesis related classification.

1.2 Related Work

The literature study for this thesis work is divided into two parts, 3D point cloud segmentation methods and online reconstruction method using point cloud segmentation. Firstly, available point cloud segmentation methods are studied from various research works, and then online-based autonomous 3D point cloud surface reconstruction by different research approaches are presented. In online-based 3D point cloud reconstruction, there are various applications such as ground and plane reconstruction, interior map generation for a building and object-centric reconstruction. Since coverage path planning is a common problem in any exploration task, the literature study was carried out irrespective of applications mentioned above.

1.2.1 Point Cloud Segmentation

A point cloud is a digital representation of any light reflecting surface. Point cloud segmentation is one of the important preprocessing steps in point cloud processing.

In point cloud segmentation, the point cloud is sub-divided into several regions to ease the analysis in micro-level. The extracted point cloud using any of the above mention methods produce a sparse point cloud with varying density. Many methods are available in point cloud segmentation; they are, edge-based method, region growing method and hybrid method. Edge-based point cloud segmentation method finds the edges of a point cloud model. A drastic change in the point cloud intensity is used to determine the edges. B. Bhanu, S. Lee, C. Ho, and T. Henderson in [10] developed a gradient-based edge segmentation algorithm where the unit normal vector on the surface is used to detect the edge points. X. Y. Jiang, U. Meier, and H. Bunke in [11] proposed a fast segmentation method for range images of both planar and curved surface. One serious disadvantage of edge-based segmentation is that it is suitable only for a dense point cloud. The region growing point cloud segmentation is a classical method of point cloud segmentation. This method is subdivided into 2D and 3D region growing segmentation methods. As our thesis work is about the reconstruction of the 3D point cloud model, literature related to 3D points segmentation in region growing method is studied. A region growing method is the process where a seed point is considered, and the region grows from there based on similar surface normal. The region growing method for 3D point clouds are further subdivided into 1) single-point approach, 2) voxel approach and 3) hybrid units. In Single point approach, X.Ning, X. Zhang, Y. Wang, and M. Jaeger in [12] developed a strategy to extract objects in 3D urban scenes for object recognition and reconstruction where the single-point approach was adopted and applied a normal vector criterion to include each point with the seed point. Even though the single point approach resulted in high accuracy, it is a time-consuming process for the vast point cloud. This problem is eradicated with the voxel approach, where a group of 3D points and its properties are considered for region growing process. This has improved the efficiency of the reconstruction process. J.-E. Deschaud and F. Goulette in [13] developed a voxel-based region growing algorithm. Vo, Anh-Vu & Truong-Hong in [14] developed an adaptive octree based region growing algorithm for fast surface patch segmentation by incrementally grouping adjacent voxels with a similar saliency feature. The hybrid method of point cloud segmentation gave the best result in terms of accuracy and efficiency, Z. Dong, B. Yang, P. Hu, and S. Scherer in [15] utilized a hybrid region growing algorithm, based on both single point and super voxels region growing method, to realize coarse segmentation. The region growing method performs better with a sparse point cloud model, but the impact of noise and outliers in point cloud processing is one identified drawback. Thus, a proper outlier removal technique should be followed before initiating region growing point cloud segmentation.

1.2.2 Online coverage path planning

Among several autonomous 3D surface reconstruction methods, it is essential to discuss the simple helical approach, proposed by Cheng et al. (2008) in [16] and Mansouri et al. (2018) in [17]. Here, the UAVs are made to fly in a helical path around the surface of the volume of interest such that the field-of-view (FOV) always faces the object's surface as shown in Figure 1.6 (Left). The drawback in this approach is that the measurement of the Volume-of-interest (VOI) must be known before the start of trajectory to decide the number of rotation and circumference of helical rotation, it neither avoids the overlap of FOV nor covers the uncovered region along the helical path in a single trajectory plan. In online Coverage path planning method, past works shall be divided into two different types, 1) Search-Based Method and 2) Occlusion-Based method. Work done by Connolly et al. (1985) in [18] proposes two algorithms to find out the "Next Best View" (NBV) through Occlusion-based method attains complete coverage. Both these algorithms segment the search space using octree (hierarchical tree structure of cubical volume). In the first method, the planetarium algorithm, where an evenly segmented sphere encloses the VOI, samples each segment as occupied or unseen area. The segments in the sphere having more number of unseen voxels are considered as the NBV. In the second method, the normal algorithm, where occupied area having more number of empty areas as neighbours are considered for NBV to complete the coverage. The NBV based reconstruction method and helical trajectory reconstruction method is depicted in Figure 1.6 (Right).

A similar kind of NBV search-based method devised by Vasquez-Gomez et al. (2014) in [3] proposes a coverage algorithm to compute NBV points to reconstruct an object using a fixed end arm robot. A utility function is built to performs ray tracing and check if the following constraints are met. New information constraint, positioning constraint, sensing constraint and registration constraint. After ray tracing, the evaluated viewpoints are again checked for position error. Finally, one single NBV is reached that satisfies all the constraints considered. Again, similar to the method discussed previously, 3D volumetric segmentation (Voxel) is used to store information about the regions. One main drawback in this algorithm is that it considers many constraints for predicting NBV that ultimately increases computation cost. L.M.Wong. (1999) in [19] came up with a new approach in NBV sensor positioning algorithm for 3D modelling of an object. An objective function is derived, that measures the quality of unknown information from each viewpoint. As explained in the previous method, the object is spatially segmented using 3D voxels and labelled them as occupied, empty and unseen voxels. He developed three different methods, Optimal method, surface-normal method and Adaptive method. In the optimal method, viewpoints are placed uniformly around the object. In the surface-normal method, summing the surface normal of the visible unknown voxel faces that point in each axis direction. The adaptive method is a mix of both optimal and surface-normal methods. The optimal method consumes more time and computationally expensive as it produces many viewpoints, whereas the surface-normal method, excludes self-occlusions of the object. The third type gives the best result in obtaining NBV. The main drawback in this method is computation time. There is no space in this method to improve the time of reconstruction.

P. S. Blaer and P. K. Allen in [20] developed an algorithm which consists of two different steps. First, a random flight to generate an initial set of point-cloud. For maximum coverage, a map of the environment is taken as input. Then in the second flight, the gaps and holes in the occupancy grid created during the random flight are investigated by flying to NBV. These NBV points are derived based on identifying the empty boundary voxels. The empty boundary voxels are the voxels which have more number of empty voxels as neighbours. The view planning algorithms presented in this manuscript eliminate unnecessary scans and reduce the time needed for detailed modelling.

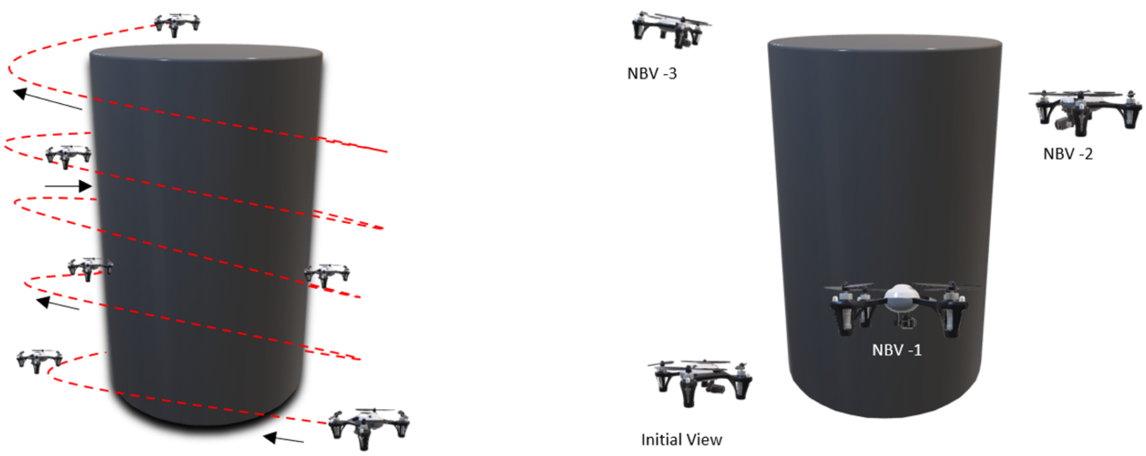


Figure 1.6: UAV flight in a helical path(Left) NBV flight method(Right) for reconstruction.

The literature study listed above motivates our thesis work in establishing complete autonomy in the reconstruction process of the partially known object. Algorithms presented in the literature study regarding point cloud segmentation reveals that voxel-based region growing method using octree data structure performs with less computational effort than other methods discussed. A new horizon of detecting point cloud edges in the sparse point cloud using the region growing method is proposed in this thesis work. In the literature study of online coverage path planning methods, there are several approaches in common. They are, data acquisition accomplished through the sensor (cameras), partially known volume of interest (at least one measurement of the object is required), spatial segmentation of point clouds (voxel grid) to decide the NBV using unique utility function specific to their work. Efforts on improving the reconstruction process in terms of time and quality are made by improvising these utility function which determines NBV. Utility function considers one or more phenomenon to estimate the best viewpoint. For example, constraints in motion, location of voxels, number of neighbouring empty voxels etc. Also, every research work has a set of criteria of to decide the efficiency of reconstruction such as completing the reconstruction process in a better time, identifying and covering the occluded regions in the object, completion of reconstruction process with least number of NBVs. Finding a novel solution to satisfy the criteria mentioned above to get better results is the key research area of this thesis work.

1.3 Method Overview

First, an initial set of point-cloud is extracted by traversing the UAV through a predefined arbitrary path in a polar-cylindrical coordinate system that completely encloses the volume of interest. Figure 1.7 shows the vertical and helical path as the initial trajectory. The latter trajectory improves reconstruction time than the former as it covers a larger surface area of the object during the initial flight. An explanation about how initial trajectory path impacts the reconstruction time is explained in Section 3.2. Point cloud extraction from surface features, localization of UAV and point cloud registration tasks are taken care of by SLAM system. Then, the extracted initial set of point cloud during arbitrary helical trajectory undergoes a refinement procedure, where the outliers are removed using crop box filter. As discussed in the literature study on point cloud segmentation, region growing method is vulnerable to outliers. Thus removing the outliers increases the efficiency of subsequent segmentation process. After this, the refined SLAM generated point cloud is spatially segmented using octree-based voxelization method. An explanation about the voxel map representation of point cloud is given in section spatial voxelization.

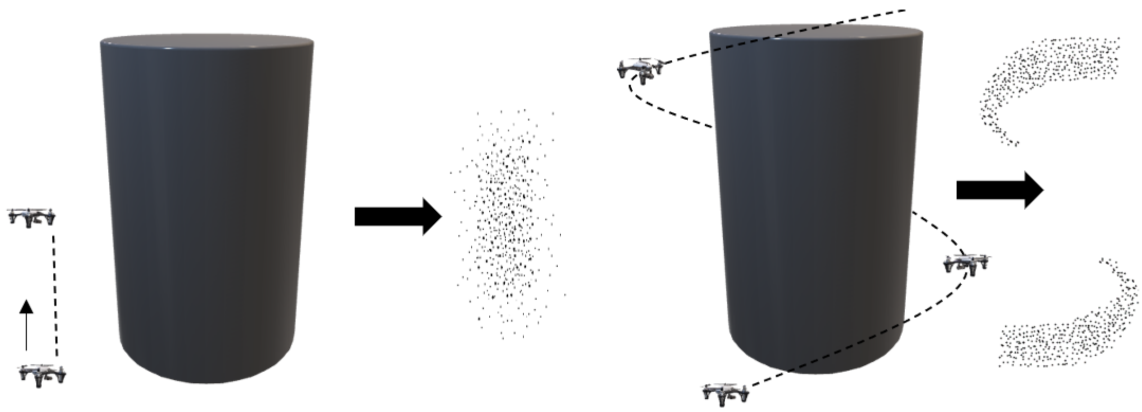


Figure 1.7: Different methods of implementing initial arbitrary trajectory around an object of interest. Vertical lift(Left) and Helical trajectory(Right).

The generated voxel grids are again refined to eliminate the voxels at dense point cloud regions. The refinement process makes use of the number of points per voxel criterion to eliminate the unwanted voxels. The list of existing voxels that are located along the boundaries of the existing point cloud is carried over to the subsequent steps. Then, the centroid coordinate of each voxel is calculated and translated away from the object's surface along the surface normal vector of neighbouring points, such that the new position lies on the polar-coordinate system. These relocated positions become the next best view(NBV) points or otherwise, way-points for the UAV during exploration flight. Then the UAV takes trajectory to these new view-points one after other while simultaneously performing feature point extraction. Once after covering the first set of viewpoints, the newly generated point cloud is registered with the existing point cloud using SLAM registration module.

The entire process continues until the termination condition is satisfied. The termination condition is satisfied if the number of feature point extraction slows down. The rate of increase in the number of feature point gradually saturates during the exploration flight. This behaviour was experimentally verified using a double helix trajectory around the object. The experiment results are presented in Section 5.4. Based on this experiment result, this termination condition is incorporated in the algorithm. The proposed algorithm is shown as a flowchart in Figure 1.8.

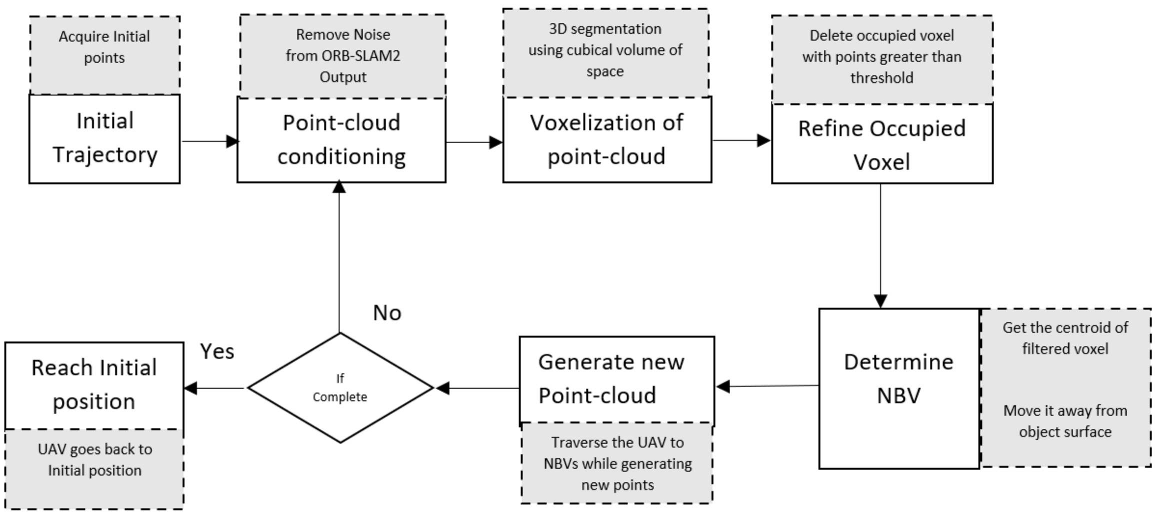


Figure 1.8: Flowchart explaining the proposed reconstruction approach.

1.4 Motivation for Proposed Method

Determining NBV though number of points per voxel criterion was derived based on the inference made by P.D. (Patrick) Radl in 2019 [9] from ORB-SLAM 2 evaluation , where it shows that as the distance between the object of interest and the UAV increases, the number of generated feature points reduces. The generated map points possess less number of points along the edges or boundaries of that set of points. This inference took the thesis work toward analysing the number of points in the segmented voxels. Figure 1.9 shows the schematic representation of this inference.

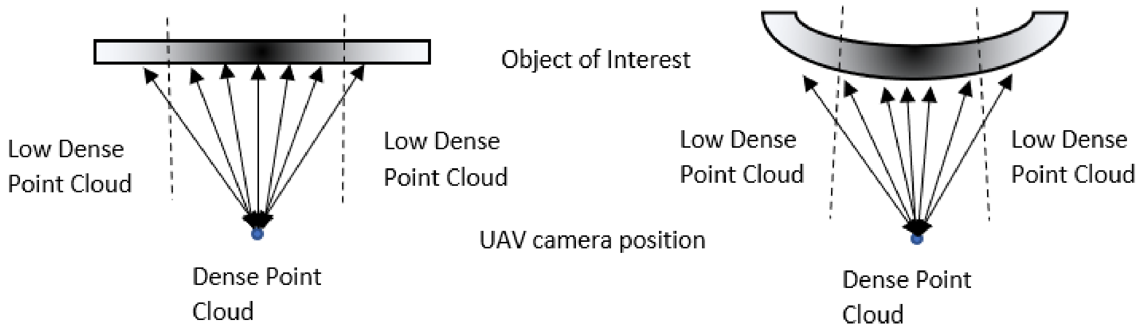


Figure 1.9: The dense and non-dense point cloud regions of a planar and cylindrical surface created by ORBSLAM2.

1.5 Problem Formulation and Research Questions

1.5.1 Research Objective

The main objective of this thesis work is to develop a methodology for UAVs that can autonomously perform reconstruction of an unknown 3D object through online (non-model based) coverage path planning approach. As a result, we obtain a complete digital representation (point cloud) of an object.

1.5.2 Research Questions

A detailed study on various research works reveals that the non-model based autonomous reconstruction of a 3D object consists of following iterative steps. Data acquisition through sensors (camera/LiDAR), feature point extraction, if the camera is used for data acquisition, spatial segmentation of point cloud to determine "Next Best View" (NBV) for progressive exploration and registration of new sets of point-cloud. Among these, deriving the next best view to exploring the uncovered regions of an object is the key area of this research work.

In our proposed method, as explained above, the number of points per voxel criterion is considered as a piece of key information to find out the next best viewpoints to improve point cloud better. The performance of proposed reconstruction method is evaluated based on quality of reconstruction and time taken to complete the reconstruction. The quality of the reconstruction process shall be concluded with two factors. One, completeness of the point cloud model.

Two, the minimum registration error. The completeness of the reconstruction process is ensured using the Iterative closest point approach and the number of feature points extracted. The minimum registration error is validated by calculating the root mean square(RMS) distance between the ground truth trajectory and the estimated camera trajectory. If the RMS distance between ground truth and the estimated camera trajectory is high, then the quality of the point cloud is reduced. The results obtained through a helical trajectory is kept as the benchmark solution for validation. The results ideally contain the number of points generated and the time of reconstruction.

And, the time of reconstruction is compared with the helical method reconstruction time. To the best of my knowledge, this approach of determining NBV was not attempted so far in determining the best viewpoints for reconstruction.This led to the following research question.

Q.1 How does the proposed methodology for 3D point cloud model reconstruction perform better in terms of time and quality than the helical trajectory approach?

As stated above reconstruction performance depends on time of reconstruction. Lesser the time better is the performance. In our proposed approach, there are various opportunities to reduce the reconstruction time. Some of these options are varying the voxel resolution and number of points (NoP) threshold limits, re positioning the NBVs, and changing the orientation of UAV. Some times while adapting to these measures, quality of reconstruction could be affected.

Q.2 How does experiments conducted to minimize the reconstruction time like tuning the voxel resolution, threshold condition, relocating NBVs, and varying UAV's pitch affects the quality of reconstruction?

As the online reconstruction approach involves exploration of an unknown objects in unknown surroundings, it is impossible to accurately estimate the time required for reconstruction. Fixing a random time limit may result in incompletion of the intended task. On the other hand, giving reconstruction time low priority might also lead to incompletion of reconstruction due to a shortage of battery capacity in an extended period of the flight. So a termination condition needs to be provided to end the reconstruction process. In our proposed method, saturation of the number of feature point is considered as the termination condition for the reconstruction process.

Q.3 How effectively does the termination condition used in our approach determine the end of the reconstruction process?

All these research questions are investigated through simulated experiments and evaluated with realistic parameters.

1.6 Report organization

Remaining part of this report is structured in the following manner. Chapter 2 discusses the background knowledge required for the reader to get in track with the work done in this thesis work. It consists of the concepts behind point cloud generation, an open-source SLAM system, octree based voxelization of point-cloud using PCL library, iterative closest point approach which is used to determine the fitness score for the reconstructed point cloud. Chapter 3 explains the proposed algorithm in detail with each step in the reconstruction process is explained with the pseudo code. Chapter 4 presents the efficiency improvement methods used in the algorithm to fasten the reconstruction process. Chapter 5 describes different types of the evaluation carried out in the thesis work. The evaluation process includes evaluation of simulation specification, evaluation of the proposed method, evaluation of performance improvement techniques and evaluation of termination condition. Chapter 6 discusses the final conclusion, where the research questions are answered with the evaluation results and also provides the scope for future work to improve the proposed reconstruction method.

Chapter 2

Background

In this chapter, a detailed explanation of the relevant background information is discussed. Following topics are covered in this chapter. The concept behind feature point extraction, point cloud generation through ORB-SLAM2 and its subcomponents, octree based segmentation of acquired point cloud model, and the working of iterative closest point approach which is used to verify the completeness of reconstruction.

2.1 Surface point generation

Point clouds are the set of discrete points that are generated from the features on the surface of any object or thing that corresponds to a coordinate system. In this thesis work, the final output of the object representation will be in the form of point cloud. Therefore, it is necessary to know the concept behind point-cloud generation. As mentioned earlier in this report, there are several ways of obtaining a point-cloud model. Our goal is to reconstruct a 3D object using a stereo camera. A 3D point extracted using the stereo-vision camera is represented as P(x,y,z) whereas in a regular monocular-vision camera a point in an image is represented as P(x,y). A stereo-vision camera consists of two image sensors, left one and a right one respectively. Triangulation method is used to extract a point from the 3D space. This concept behind the point-cloud generation using the stereo-vision camera is discussed in the following section.

2.1.1 Point Generation by Triangulation Method

A 3D point is determined in a given space using triangulation method given two different projection of a scene. As mentioned before, a stereo-vision camera provides two different projections through the right and left image sensors. The intersection of two known rays is sufficient to estimate the point coordinate. To determine the point coordinate using the intersection rays, the following information is required. The parallel optical axis of both the cameras(unparalleled axis involves a different triangulation approach to find point position), geometrical arrangements and internal parameters of the stereo cameras. The triangulation method is depicted in

Figure 2.1.

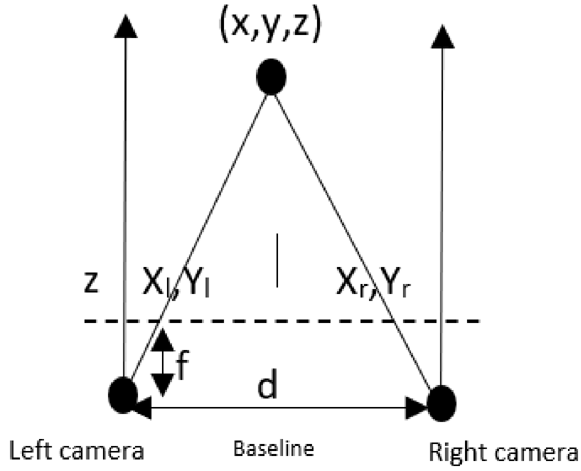


Figure 2.1: Depicts the stereo vision process using triangulation method.

The distance between two camera centers is called baseline. Let the baseline be perpendicular to the optical axes of the cameras and parallel to the x-axis. The distance between cameras is d and cameras have equal focal length f. Z is the depth of the point in a three-dimensional world. The point (x,y,z) have x and y coordinate in each camera, respectively as (X_l, Y_l) and (X_r, Y_r). From similar triangles in the figure implies

$$Z = \frac{d_f}{X_l - X_r}$$

Where $X_l - X_r$ is the disparity. A disparity map is a representation of depth information which is calculated using two images obtained from the stereo-vision camera. Thus, the disparities of the corresponding image points are used to retrieve depth at various scenes. Also, x and y are calculated using the below equations

$$X = \frac{d(X_l + X_r)}{2(X_l - X_r)}$$

$$Y = \frac{d(Y_l + Y_r)}{2(Y_l - Y_r)}$$

Now, the generated points have to be grouped in their respective position, which in turn represents the volume of interest. The following section explains the system which performs this process.

2.2 SLAM

The purpose of Simultaneous Localization and Mapping (SLAM) algorithms is to build a map of an unknown environment and determine the position of the sensor the perceives data about the operating environment. It is used in various applications mainly focused on robotic navigation, driver-less cars and virtual reality. SLAM is also used as the basic function in many path planning algorithms. The basic architecture of SLAM algorithms consists of four components: perception, loop closing, mapping and optimization. There are various open-source SLAM systems available. These SLAM algorithms can be broadly classified into a feature-based and direct method. Feature-based method completely relies on the surface features for SLAM operation. Whereas direct method exploits all the information for its operation.

Developed by Mur-Artal and Tardos (2017) in [2] , ORB-SLAM2 is the successor of ORB-SLAM, with an added advantage of using it in stereo-vision camera in addition to the monocular-vision camera. Oriented FAST and Rotated BRIEF (ORB) is a fast, robust local feature detector which is based on the Features from Accelerated Segment Test (FAST) key-point detector and the visual descriptor Binary Robust Independent Elementary Features (BRIEF). It replaces SIFT detectors which are generally used to extract features. Scale Invariant Feature Transform (SIFT) is highly computationally expensive. Thus ORB feature extraction technique has reduced computational load to CPU in real-time. The working of ORB feature extraction is described by Ethan Rublee and Vincent Rabaud in 2011 [21]. ORB-SLAM2 is divided into three threads, tracking, local mapping and loop closing. The functional block diagram representation is shown in Figure 2.2.

2.2.1 Tracking

ORB-SLAM2 processes every input frame of stereo output and determines the distinguishable feature points which are then converted to map points. Feature extraction is carried out using the ORB feature detection algorithm by Rublee et al. (2011) [21].

Camera pose is calculated from the previous frame using the constant velocity model, and feature matching is performed between the frames. If the tracking is lost, the current frame is compared to key-frames to match features and thereby relocalise. If in both cases enough features are found the current pose of the camera is predicted. With these initial set of features and camera pose, the map is projected to search for map point in correspondence of features. Computation load is reduced by selecting points from only the related key-frame. Related key-frames are key-frames that should have map points in common with the current frame and their neighbours. Neighbouring key-frames are determined by their shared map points with each other. Based on the corresponding points, which are also in the frame visible, the camera pose gets optimized and updated.

Determining new key-frame is one of the core function of tracking. According to Mur-Artal et al. (2015) as in [2], the current frame must satisfy the following criteria.

- Re localization and key-frame insertion must be done at least after every 20 frames.
- At least 50 points are tracked.
- At least 90% of the points of the key-frame with the highest map point correspondence are tracked.

One exception for all these exits. Using the stereoscopic version of the algorithm in this research, the initial frame is used as initial key-frame, as Mur-Artal and Tardos (2017) point out. At this point, the camera is also set to the origin, and an initial map is created from detected features.

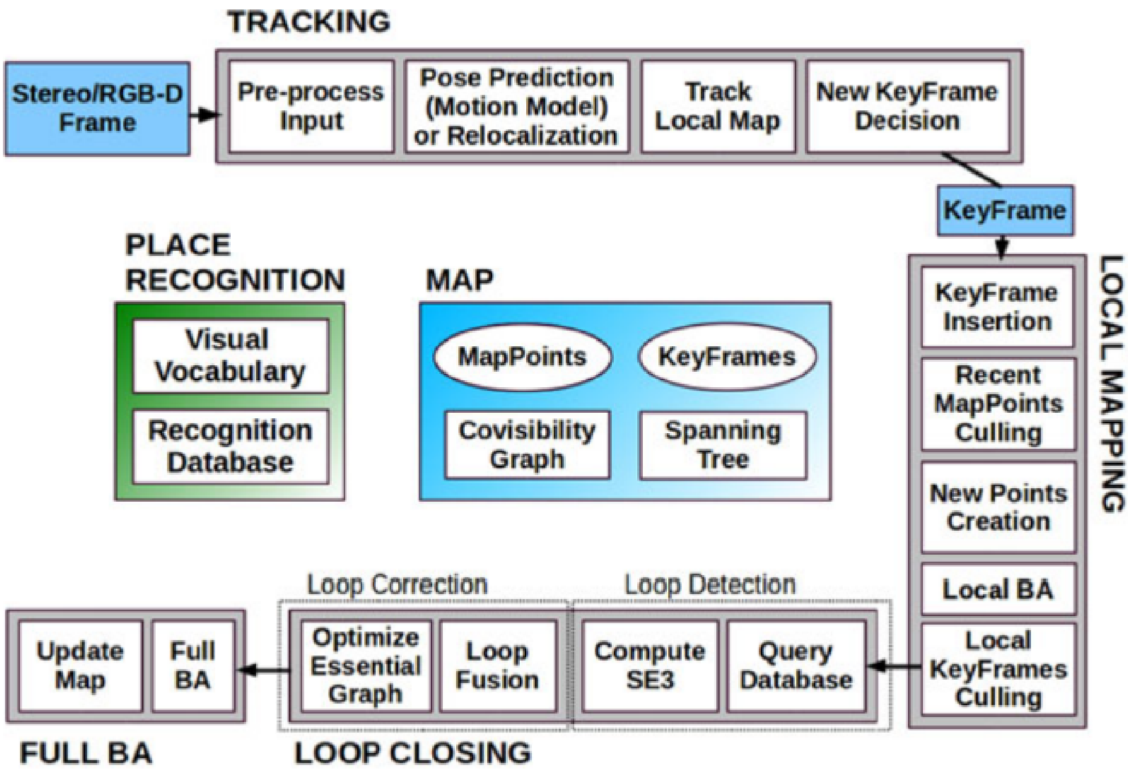


Figure 2.2: Structure of ORB-SLAM2 by Mur-Artal and Tardos (2017), showing the three main threads [2].

2.2.2 Local Mapping

Local Mapping is performed at every new key-frame. It consists of five different processes such as Key Frame Insertion, Recent map points culling, New map points creation, Local bundle adjustment (BA) and Local key-frame culling.

In the **Key Frame Insertion** step, a co-visibility graph is updated where a new node is updated in the new key-frame. The new key-frame is linked with other key-frames having common points. Then a bag of words is computed to represent the key-frame that helps in the triangulation of new points.

In **Recent Map Points Culling**, the map points are tested to be retained in the next three key-frames after creation, to ensure that it was not wrongly triangulated. The points must fulfil the following two conditions.

1. The tracking must find the point in more than 25% of the frames in which it is predicted to be visible.
2. If more than one key-frame has passed from map point creation, it must be observed from at least three key-frames.

Once a map point passes this test, it can only be removed if at any time it is observed from less than three key-frames. This can happen when key-frames are culled and when local bundle adjustment discards outlier observations. This policy makes our map contain very few outliers.

In **New Map Points Creation**, new map points are created through triangulation method as stated in section 2.1.1. For each unmatched ORB in new key-frame, a match with another unmatched point in other key-frames will be searched. This is done by “Bags of Words Place Recognition” method as described by Gálvez-López et al. (2012) [22].

In **Local Bundle Adjustment**, currently processed key-frames are optimised. All key-frames are connected to the co-visibility graph. All other key-frames that see those points but are not connected to the currently processed key-frame are included in the optimization but remain fixed. Observations that are marked as outliers are discarded at the middle and at the end of the optimization. In Local Key-frame Culling, the local Mapping tries to detect redundant key-frames and delete them.

2.2.3 Loop Closing

ORB-SLAM2 provides a loop closing feature, which targets to reduce the accumulated error during the observation. When reaching the same pose, this drift in the map should be possible to estimate. Hence, the last key-frame processed by the local Mapping is compared with a database to determine if the scene was previously observed. Between the detected loop key-frame and the current key-frame, a map point matching is applied to determine the error accumulated in the loop to construct a “similarity transformation”. In a first step, this transformation is applied to the current frame and its neighbours to fuse matched map points. Finally, all the key-frames within the loop are getting transformed to remove the drift error. Loop closing also corrects the map points covered by those key-frames.

In ORB-SLAM2 a full BA is invoked after closing a loop. As it is computationally very costly and to not avoid detecting new loops, it is running in a separate thread. Full BA distinguishes from the previously mentioned local BA that it is optimizing all key-frames and all map points. When it is finished, the map has to be updated with the output of the full BA. Meanwhile, newly introduced key-frames and their map points are transformed by propagating the correction, which was applied to the optimized ones.

2.3 Spatial Voxelization of Point Cloud

Analysing a point cloud model of a large structure or object as a single entity does not help in localized inspection for reconstruction. Hence a modular analysis of point cloud is required to detect the incomplete regions in the surface 3D point cloud model. Voxelization is a process in which a 3D cubical space similar to pixels in the 2D bitmap overlays the input point cloud, as shown in Figure 2.3 (Right). The trace ability of voxel information is established through an octree based spatial segmentation. It is a continuous grip of connected voxels in the cartesian space. In the octree data structure, space decomposition is performed congruently. Each node is split into eight children node of the same dimension and is labelled based on the occupancy of points in it. An occupied node contains map points, and an empty node does not have map points. The voxelization initially fits in minimum cubical bounding box enclosing entire point cloud which is called level 0. Then this cube recursively sub-divides into eight children node until the minimum resolution criteria are met. Each subdivision marks a level of nodes. Each node in the octree data structure stores several salient features. They are coordinates of 3D points, the number of points, surface normal of point, the label of the node(occupied or empty)ad bounding limits of each node. These features are then used application-specific. Figure 2.3 (Left) shows the pictorial representation of octree data structure.

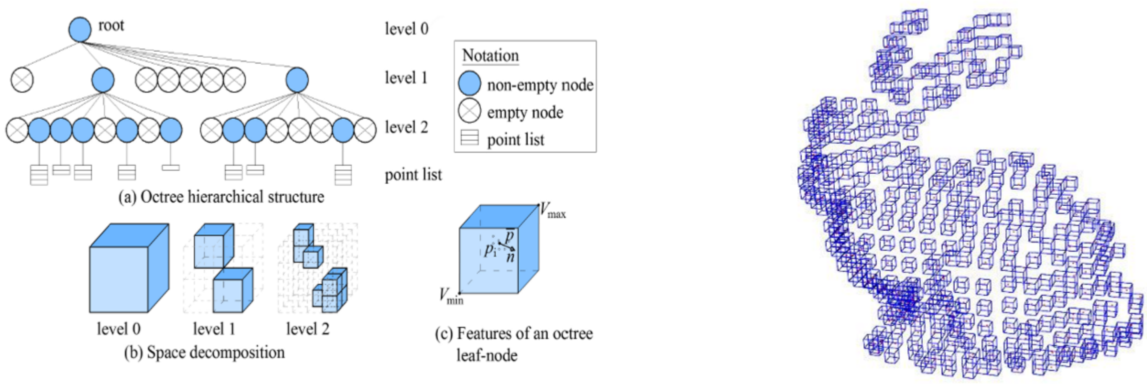


Figure 2.3: An Octree data structure [3](Left) and voxelization of stanford bunny rabbit [4](Right).

2.4 Iterative Closest Point

The Iterative Closest Point algorithm was introduced by Besl and McKay in [23] to aligning two 3D point clouds by minimizing the distance between them using geometric transformations (rotations and translations). The ICP algorithm has two steps:

1. Determine the correspondence pairs (K) between two data sets P and Q. Where $P = p_1, p_2, p_3, ..., p_n$,$p_i \in R^3$ and $Q = q_1, q_2, q_3, ..., q_n$, $q_j \in R^3$ with current transformation matrix (T). The goal is to find for each point in P its closest point in Q .
2. The second step is to update the transformation (T) (rotation and translation) in order to minimize the objective function E(T) over the correspondence pairs (K).

These two steps are repeated until the error is below a given threshold or until the maximum number of iterations is reached. A cost function is derived using least mean square approach. The algorithm considers each point cloud consist of a set of distance measurements.

$$E(T) = \sum_{(p,q)\in K} \|p - Tq\|^2$$

The ICP algorithm establishes a transformation parameters iteratively between the corresponding points in both point clouds so that the mean square distance is minimized. ICP yields a metric called fitness score which describes the overlapping area between the two point clouds and is defined by the formula

$$F = \frac{N_k}{N_p}$$

where N_k is the number of correspondences and N_p is the number of points in the target point cloud.

Chapter 3

Proposed Method

Our proposed algorithm to reconstruct the point-cloud model of a 3D object using UAVs consists of two-staged flight initial arbitrary and exploration flight. The exploration flight is the subsequent step of arbitrary flight. During the arbitrary flight, a pre-planned trajectory around the object of interest is executed to generate an initial point-cloud model of the surface. This initial flight plan does not establish a complete coverage around the object. Instead, it initiates the ORB-SLAM2 system for pose estimation and tracking. The initial arbitrary flight plays an essential role in reducing the time of reconstruction by covering the maximum surface area without any computation. Thus choosing an optimal initial trajectory path is vital. Among the two different initial flight path discussed in the method overview section, the helical method covers a maximum area of on the object's surface. The explanation for this choice is presented in the Section 3.2.

The second flight is an exploratory flight in which the UAV flies to the list of refined NBVs determined by our algorithm. The UAV on flying to all the determined NBVs, ORB-SLAM2 system generates the feature points and builds the point cloud model. The exploratory flight ends only when the condition for reconstruction completion is satisfied. The two-staged flight method was inspired from work done by Tobias Koch and Marco Körner 2019 as in [24], where they have implemented two stages of scanning using UAV flight planning for image-based 3D reconstruction. The initial set of point-cloud extracted through arbitrary flight undergoes filtration process to remove noises and then spatially segmented using octree-based volumetric segmentation method explained in Section 2.3. A vector of occupied voxels having feature points less than a predefined threshold limit is pushed to the subsequent step, and other voxels that do not satisfy the threshold limit are discarded from the process at this stage. Then the centroids of these filtered voxels are calculated and translated along the surface normal, away from the object such that they lie on the polar coordinate system. These translated centroid point on the polar coordinate surface are the NBV points which also acts as the waypoints for UAV trajectory, preventing the vehicle from colliding itself from the volume of interest. Once the NBV points are determined, the explorative flight begins.

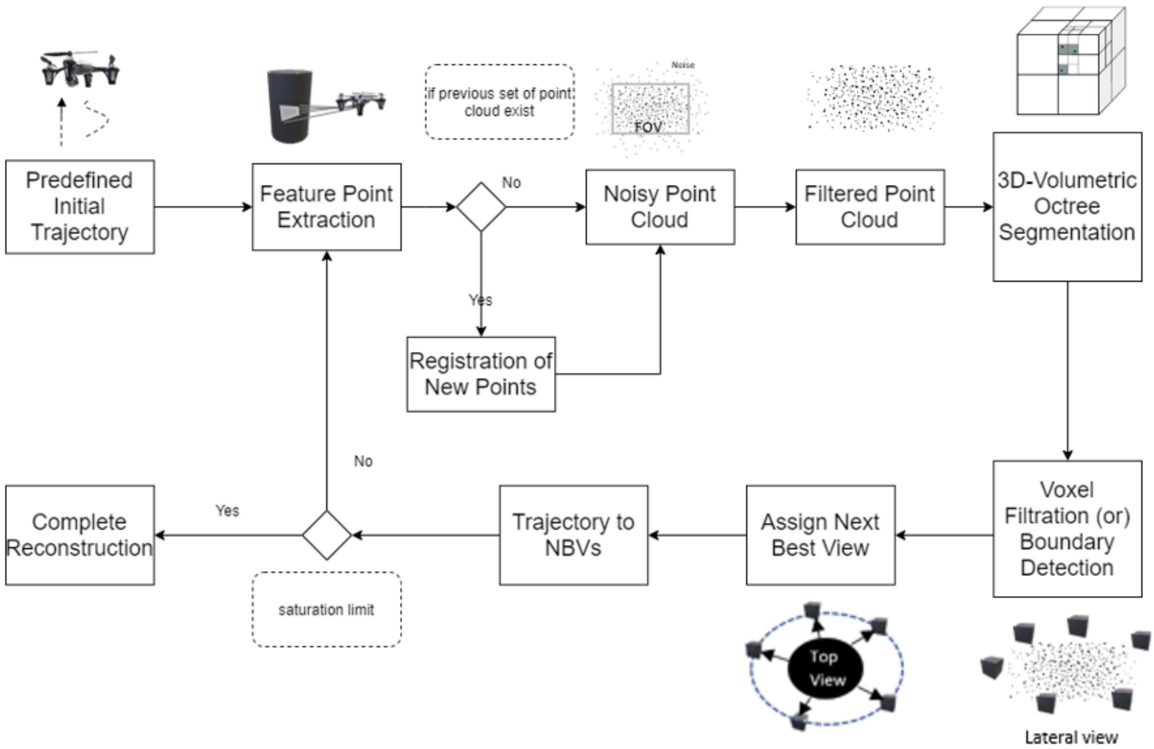


Figure 3.1: The detailed block diagram of proposed algorithm.

The UAV executes smooth trajectory to all the NBV points, camera facing the object's surface. During the explorative flight, ORB-SLAM2 system simultaneously performs feature point generation, pose estimation, tracking, and registration of new points with the existing point-cloud. Also, the point-cloud processing such as outlier removal, spatial volumetric segmentation and NBV generation, which are common to both stages of flights, continues until the reconstruction process is complete.The reconstruction process is said to be complete if the number of points extracted from the texture is saturated within the predefined range. For a given object with a given texture, the number of feature points does not increase beyond a specific limit. This observation was witnessed by conducting a double-helical trajectory experiment using the same object of interest. The experiment results are shown in Section 5.4. Thus for a given textured object, the maximum number of feature points saturates at some stage of reconstruction. If there is no vast difference in the number of points in the point-cloud, reconstruction is said to be complete. A detailed representation of proposed algorithm is presented in Figure 3.1. The following part of this proposed method section contains a detailed explanation of the UAV trajectory, initial trajectory, point cloud processing, positioning the next best viewpoints, Stabilising the NBVs, explorative trajectory to NBVs and termination condition for the reconstruction process.

3.1 UAV Trajectory

The UAV is bound to fly in a polar-cylindrical coordinate system that encloses the entire object on the x-y plane. This aids in avoiding collision with the object of interest. Unlike cartesian coordinate system where a spatial point is represented as (x,y,z), in a polar-coordinate system a spatial point is represented as (r,θ,z) where r is the distance of the point from central axis at the origin, θ is the angle between the x-axis and the line segment connecting central axis to the point and z is the height. A polar-cylinder coordinate system is pictorially presented in Figure 3.2. The radius of the polar-coordinate system is chosen as 5 meters. The reasoning for choosing this radius is discussed in Section 5.1.3.

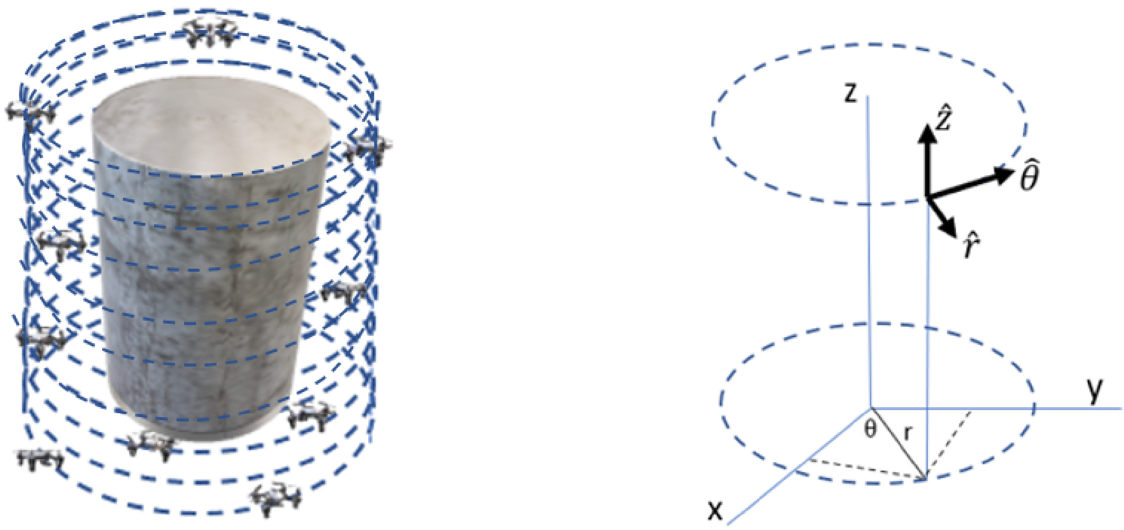


Figure 3.2: The polar cylindrical coordinate system enclosing the object of interest (Left) and the cartesian representation of polar cylindrical coordinate system.

A UAV in the polar coordinate system either follow a circular path or vertical displacement or both simultaneously. The circular trajectory followed by the UAV is defined by the equations 3.1,3.2, and the vertical trajectory is defined by equation 3.3.

$$x(t) = r * cos(\pm\omega_t) + x_0 \tag{3.1}$$

$$y(t) = r * sin(\pm\omega_t) + y_0 \tag{3.2}$$

$$z(t) = \pm(V_z t) + z_0 \tag{3.3}$$

In equation 3.1 and 3.2, r is the radius of the polar coordinate system, t is the trajectory time which corresponds to the simulation step function. The duration of a step function is denoted by Δt_s ($\Delta t_s = t_n - t_{n-1}$), ($\pm\omega$) is the angular displacement in counter-clockwise (positive) and clockwise (negative), and (x_0, y_0) is the position of the central axis of the object. In our experiment, the centre of axis lies at the origin. In equation 3.3, V_z is the velocity along z-axis both in upward$(+V_z)$ and downward $(-V_z)$ direction and z_0, the current altitude of UAV. The combination of above equations leads the UAV towards a required position in the polar coordinate system. For a helical trajectory, all the three equations are triggered simultaneously with all the parameters (r,ω,t) set before the start of trajectory.

3.2 Initial Arbitrary Trajectory

The first stage of flight is an arbitrary trajectory in the polar coordinate system. There are three possible combinations of the trajectory plan in a polar coordinate system. One, along the z-axis. Two, a circular path around the object in the x-y plane. Three, a combination of one and two. Figure 3.3 shows the pictorial representation of three possible trajectories on a polar coordinate system. Among these, the third plan results in a pure helical trajectory and covers a maximum area on the surface, provided the helical path length is equal to the diagonal length of the 2D projection of 3D cylindrical surface. If a 3D cylindrical surface is projected on the 2D surface it forms a rectangular plane where the height of the cylinder corresponds to the breadth of the rectangular plane, the circumference of the cylinder corresponds to the length of the plane and a helix path of maximum length corresponds to the diagonal of the rectangular plane.

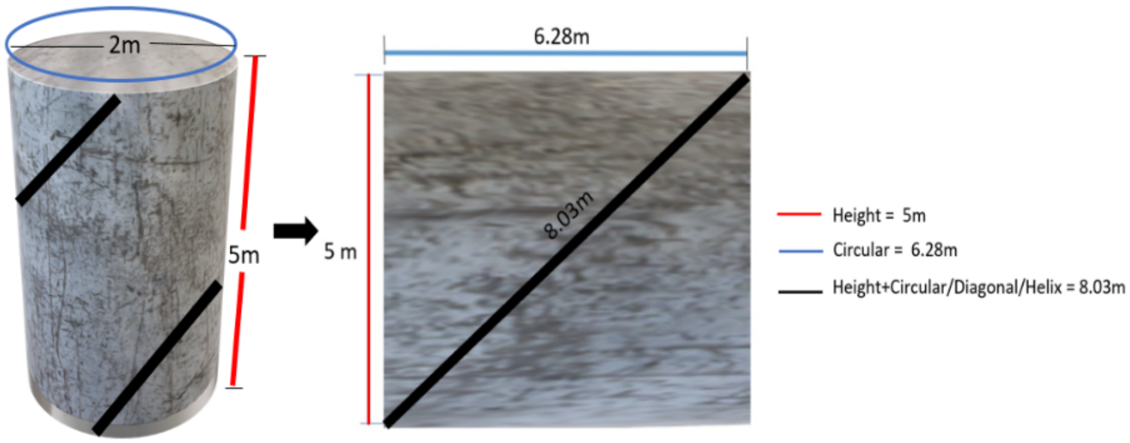


Figure 3.3: The helical trajectory around an object in polar coordinate covers more surface area than other options.

Among the three possible types of trajectories, a simple mathematical calculation reveals that the helical trajectory covers more surface area of the surface. Thus in our algorithm, the initial trajectory is chosen to be a helical trajectory. Equation of the helical trajectory is provided in Section 3.1. The pictorial representation of the helical UAV trajectory around the object and the resultant point cloud is shown Figure 3.4.

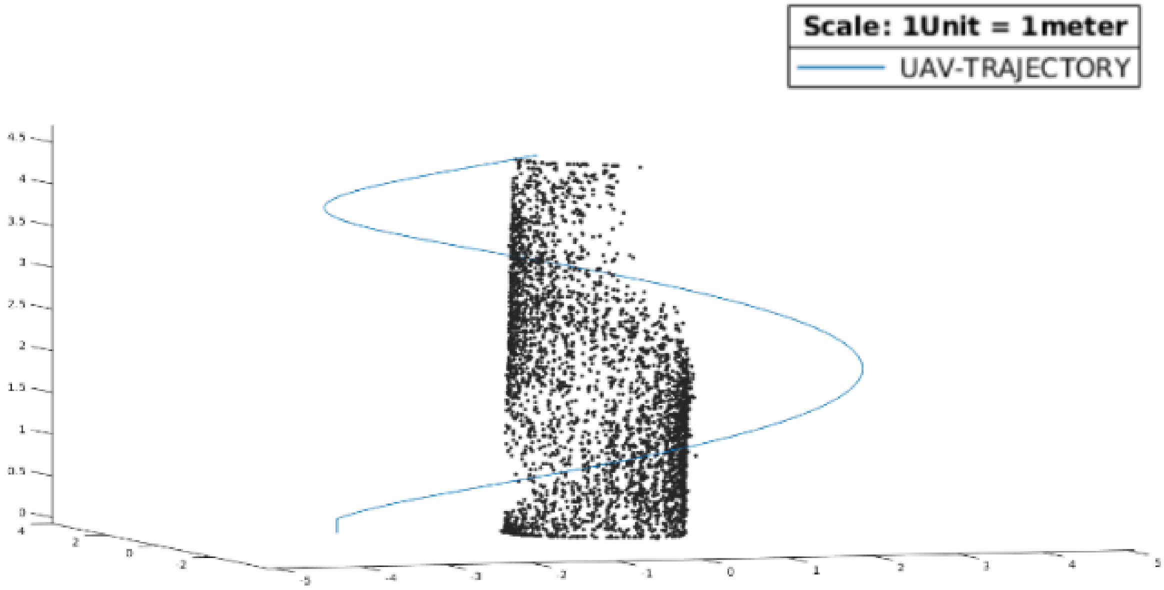


Figure 3.4: The simulation result of the helical trajectory around the object of interest.

Before initiating the helical trajectory, the UAV is made to perform a negligible amount of vertical lift to initiate the ORB-SLAM2 system which is available in Figure 3.4. This step is used to generate an initial set of map points that helps the ORB-SLAM system to initiate the localization process. The stages of flights such as vertical lift, helical trajectory and inspection trajectory are synchronized within the state machine. The pseudo code of transitions in the state machine is shown in Algorithm 1 as in Figure 3.5.

Algorithm 1: Trajectory State Transition

Data: GUI input to initiate the process, G_{INIT} = [ON or OFF]. The current state of UAV denoted as U_{STATE}={U_{INIT}, U_{HELIX}, U_{WAIT}, U_{REST} and U_{INSPEC}}. The current state of UAV, S_{CUR}. The previous state of UAV, S_{PREV}. Initial height to be lifted, H_{INIT}. Height of Helix flight, H_{HELIX}. Current Height of UAV, H_{CUR}. Status of UAV denoted as S_{UAV} = {Init_done, Helix_done, Inspec_done, Recons_done}
Result: UAV status, S_{UAV}.
Initialization: $G_{INIT} \leftarrow$ OFF , $U_{STATE} \leftarrow U_{INIT}$
function State_Transition(T)

```
begin
  if G_INIT ← ON then
    State_Machine();
  else
    U_STATE ← U_REST;
  function State_Machine()
    if ( (U_STATE = U_INIT)&&( H_CUR ≠ H_INIT))  then
      Fly_to_InitHeight();                          //Equation(3.3)
      S_UAV ← Init_Inprogress;
    else
      U_STATE ← U_WAIT ;
      S_PREV ← U_INIT;
      S_UAV ← Init_done;
      return S_UAV;
    if ((U_STATE = U_HELIX)&&( H_CUR ≠ H_HELIX)) then
      Fly_Helix();                                  //Equation (3.1)&(3.2)
      S_UAV ← Helix_Inprogress;
    else
      U_STATE ← U_WAIT ;
      S_PREV ← U_HELIX;
      S_UAV ← Helix_done;
      return S_UAV;
    if (U_STATE = U_INSPEC) then
      Fly_Inspec();                                 //Equation(3.1),(3.2)&(3.3)
      S_UAV ← Inspec_Inprogress;
    else
      U_STATE ← U_WAIT ;
      S_PREV ← U_INSPEC;
      S_UAV ← Inspec_done;
      return S_UAV;
    if (U_STATE = U_REST) then
      Fly_InitPos(); |
      S_UAV ← Torest_Inprogress                     //Equation(3.1),(3.2)&(3.3)
    else
      U_STATE ← U_WAIT ;
      S_PREV ← U_REST;
      S_UAV ← Recons_done;
      return S_UAV;
    if (U_STATE = U_WAIT) then
      do_nothing();
    else
      if S_PREV = U_INIT then
        U_STATE ← U_HELIX;
        S_PREV = U_WAIT;
      else if S_PREV = U_HELIX then
        U_STATE ← U_INSPEC;
        S_PREV = U_WAIT;
      else if S_PREV = U_INSPEC then
        U_STATE ← U_REST ;
        S_PREV = U_WAIT;
  end function
end
end function
```

Figure 3.5: State transitions of the UAV.

3.3 Point Cloud Processing

The object of interest is expressed in the form of discrete set of surface points P = p1,p2,p3,. . . . .pn where each point denotes a 3D spatial coordinate. Map points generated by ORB-SLAM2 contains noisy data which increases the computation load and reduces the performance in determining best viewpoints. For this reason, it is essential to remove the outliers from the raw ORB-SLAM2 output. We took advantage of the PCL open-source library for point cloud processing. PCL library consists of various open-source algorithms to work on point-clouds. Crop-box filter is one such PCL functionality where a user-defined box which filters only the data points that fall inside the frame. Other points are omitted from further processing. Octree-data structure is another functionality from the PCL library used for spatial segmentation of point clouds. The detailed information regarding the octree data structure is discussed in Section 2.3. The filtered point cloud is fed to the octree data structure for octree decomposition. Octree decomposition is executed based on the resolution criterion. In our algorithm, the octree resolution is cuboid of length 0.4 meters for the chosen object of interest. The impact of varying the octree resolution is discussed in Section 5.3.1. The segmentation of point cloud involves a grid of connected voxels (Nv = number of voxels in the octree data structure) which are denoted as V= v1,v2,v3,. . . . .vn. These voxels undergo a filtration process using one of the salient features in the octree data structure, the number of points in each voxel (Np). A threshold limit for Np is set to filter the voxels below the limit. Vthres denotes the threshold limit. The set of voxels which hold the number of voxels below Vthres are grouped as R =r1,r2,r3,. . . . .rn. These collection of voxels appears along the boundaries of the existing dense point-cloud model, as shown in Figure 3.7. Then the coordinates of these voxel centroids are determined using voxel bound limits. This piece of information is calculated using the minimum and maximum voxel bound limits that are stored in the octree data structure. These minimum and maximum limits along x,y and z-axis are denoted as v_{xmin}, v_{xmax}, v_{ymin}, v_{ymax}, v_{zmin}, v_{zmax} respectively. Centroid coordinates(v_{cx}, v_{cy}, v_{cz}) of the filtered voxels are determined using the formula 3.4.

$$v_{cx} = \frac{v_{xmin} + v_{xmax}}{2} \; v_{cy} = \frac{v_{ymin} + v_{ymax}}{2} \; v_{cz} = \frac{v_{zmin} + v_{zmax}}{2}, \tag{3.4}$$

The centroids of filtered voxels are listed as $C_i = 1, 2, ..., n = v_i \in C$, comprised of the 3D location of centroids $v_i \in R3$. The pseudo-code of the octree-based point cloud processing in the algorithm is described in Algorithm 2 as in Figure 3.6. The simulation result of determining the centroid of low-density voxels is shown in Figure 3.7. The blue points in the figure are the ORB-SLAM2 generated point cloud model, and the red points are the position of the centroid of low-density voxels in the octree data structure.

Algorithm 2:Octree-based Point-cloud processing

Data: Octree O having a list of leaf nodes $V = \{v_1, v_2, v_3, ...v_n\}$, Low Dense Voxel list $V_L = \{V_{L1}, V_{L2}, ...V_{Ln}\}$,set of input point-cloud $P=\{p_1, p_2, p_3, ...p_n\}$, Voxel density or NoP(number of points) threshold Vth, Number of point in each voxel $N=\{V_{N1}, V_{N2}, ...V_{Nn}\}$.
Result: set of coordinates (centroids of low dense voxels) $C = \{c_1, c_2, c_3, ...c_n\}$
Initialization: centroid list $C \leftarrow \emptyset$, $V_L \leftarrow \emptyset$, ;
function Voxelization(P)
  begin
    Voxelization of input point cloud $O = V(P)$
    while $O \neq \emptyset$
    for each leaf node $v_i \in O$ **do**
      if $V_{Ni} < V_{th}$ **then**
        $V_L \leftarrow$ *low dense*(v_i, O);
    for each $V_L \in O$ **do**
      $C \leftarrow$ centroid(V_{Li}); //equation(3.4)
    return C;
  end function

Figure 3.6: Extract centroids of Low Density Voxels.

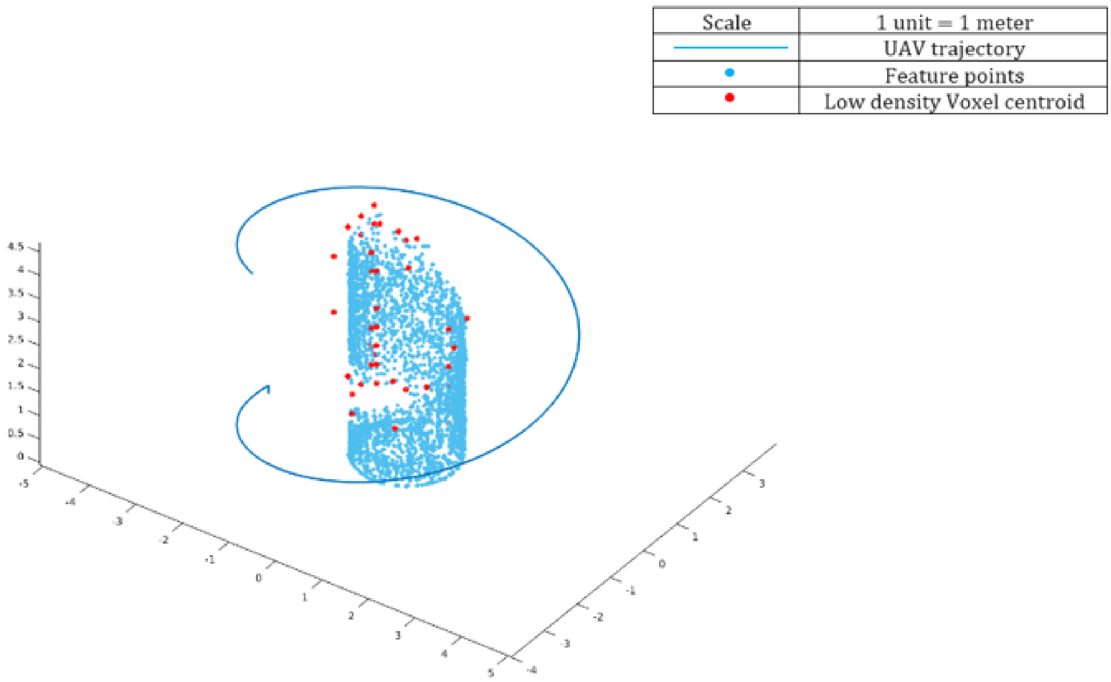


Figure 3.7: The simulation result of Low-density voxel centroids in Octree data structure.

3.4 Positioning Next Best Views (NBVs)

The centroids obtained through octree-based point cloud processing are the regions to be improved to complete the reconstruction process. The UAV in its exploration flight has to fly to these regions to generate more feature points. These centroid points are not the surface points, but they lie among the surface points. If the UAV is made to fly to these positions, it collides with the surface. Hence it is necessary to translate these centroid points away from the surface such that UAV fly to these points to generate new sets of points. In our algorithm, the UAV is restricted to fly only in the polar coordinate system to avoid the computational effort for obstacle detection. It is a viable solution to translate these centroid points on to the polar coordinate system. These new position of centroids in the polar coordinate system are the NBVs for UAV to fly during the exploratory flight. The process of translating the centroid points on the polar coordinate system is explained in Algorithm 3 as in Figure 3.8 . The simulation result of positioning the NBVs are shown in Figure 3.9.

Algorithm 3: Positioning NBVs on Polar coordinate System

Data: List of centroid coordinates to be translated, $C = \{c_1, c_2, c_3, ...c_n\}$, Distance of the centroids from origin, $D_{fc} = \{d_1 d_2, ...d_n\}$, Radius of Polar coordinate system R_p
Result: List of translated point on polar coordinate system $NBV_s = \{vp_1, vp_2, ...vp_n\}$
Initialization: $NBV_s \leftarrow \emptyset$, Distance to modify, $D_{mod} = 0$;
function Position_Centroid(C)
  begin
    for each centroid points C **do**
      get angle of centroid points made with positive x-axis, theta(c_i,o);
      get distance from origin $d_i \in D_{fc}$;
      if $d_i < R_p$ **then**
        $D_{mod} = R_p - d_i$;
        $vp_i.x = (D_{mod} + d_i) * \cos(theta)$;
        $vp_i.y = (D_{mod} + d_i) * \sin(theta)$;
        Translates to new position in X-Y plane in polar coordinate
      else
        $D_{mod} = d_i - R_p$;
        $vp_i.x = (d_i - D_{mod}) * \cos(theta)$;
        $vp_i.y = (d_i - D_{mod}) * \sin(theta)$;
        Translates to new position in X-Y plane polar coordinate
      return $NBV_s \leftarrow vp_i$;
    end
  end function

Figure 3.8: Positioning voxel centroids in polar coordinate system.

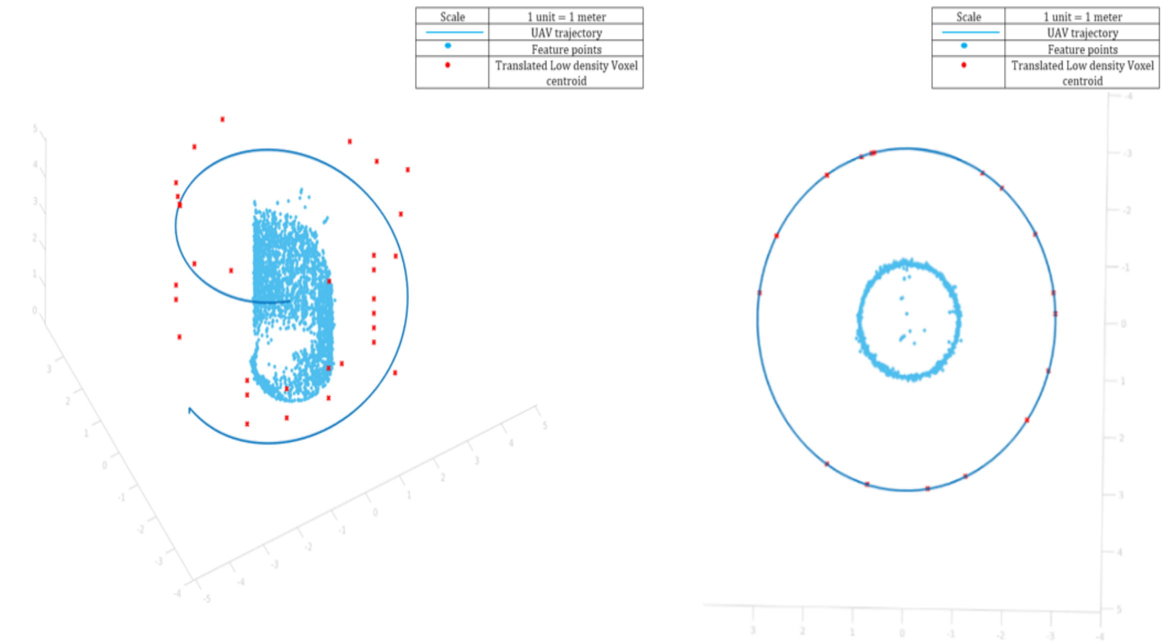


Figure 3.9: The translated centroid points towards the polar coordinate system(left) and Top view of the same (right).

3.5 Stabilize the NBVs

As mentioned earlier in the method overview, the ORB-SLAM2 system constantly generates the point cloud at the frequency 0.01s-1. This frequency corresponds to the ROS timer set to schedule the callback functions. Defining this frequency as low as possible results in the smooth simulation process. The callback functions are called once in every 0.01 seconds. The callback functions consist of the proposed algorithms(Algorithm 1-4). As a result the NBVs which are determined using the NoP threshold limit, is also called every 0.01 seconds. The UAV which performs trajectory constantly generates new point cloud resulting in the dynamic behaviour of NBVs. Because, the number of points in each voxels are constantly varying that lead the NBV list to get updated every step cycle(0.01s). This phenomenon could vanish the destination NBV even before the UAV reaches the destination point. These unstable NBV list results in the wobbling behaviour of UAV. Hence it is vital in the algorithm to hold the NBV list unchanged until UAV completes the trajectory to the destination NBV. In our proposed method the ROS node which generate NBVs are made to go sleep for a predefined amount of time, right after the end of initial arbitrary trajectory. This helps the generated NBV point to be unaltered for a predefined time limit.

At the end of exploration trajectory, the new point clouds are registered with the existing point cloud.Figure 3.10 describes the algorithm as the signal diagram.

The generated feature points after the initial trajectory, is kept on hold until UAV performs trajectory to all the five NBVs. At the same time ORB-SLAM2 generates the feature point constantly from the start of initial trajectory. In the following section, algorithm for trajectories to all the NBVs are explained.

3.6 Trajectory to NBVs

The UAV at this point in algorithm comes out of the helical trajectory stage and transfer to exploratory trajectory stage. The list of NBV's mentioned in Algorithm 2 are the new locations for UAV to fly to. The UAV initiate the trajectory to each NBVs in the list. The ORB features are extracted simultaneously during the exploration flight. New feature points after going through point cloud processing, are registered with the existing point cloud generated during the initial flight. These new set of points are again voxelated to obtain new NBVs and are added to the existing list. This iteration continues until the reconstruction process is complete. Selecting the shortest path is vital in reducing the reconstruction time. In our algorithm, it is calculated based on four quadrants x-y plane. The pseudo-code for selecting the shortest path is presented in Algorithm 4 as in Figure 3.11.

Algorithm 4: Selecting the shortest path in the polar coordinate system

---

Data: Current angle of the UAV with respect to the positive x-axis, U_{START}, Coordinates of NBVs, NBV={nbv_1,nbv_2,nbv_3,......nbv_n}. The angle of nbv_i with respect to the positive x-axis, U_{DEST}. Current UAV angle with respect to positive x-axis U_{START}.
Result: Direct of trajectory clockwise(0) or anti-clock wise(1).
Initialization: Inspection State, $U_{INSPEC} \leftarrow$True, Helix State, $U_{HELIX} \leftarrow$ False, Direction of trajectory, D
function Direction_To(NBV)
  begin
    NBVs← Position_Centroid(C);
    get $U_{DEST} \leftarrow \tan^{-1}(\frac{nbv_x}{nbv_y})$, $U_{START} \leftarrow \tan^{-1}(\frac{UAV_x}{UAV_y})$
    if ($U_{START} \neq U_{DEST}$) **then**
      $A_{DIFF} \leftarrow U_{START} - U_{DEST}$;
      if ((A_{DIFF} > 0)&&(A_{DIFF} < M_PI))|| (A_{DIFF} > - M_PI) **then**
        D ← clockwise;
      else if ((A_{DIFF} < 0) && (A_{DIFF} < -M_PI))||(A_{DIFF} > M_PI) **then**
        D ← anti-clock wise;
      return D ;
    end
  end function

Figure 3.11: Choosing the shortest path to NBV in Polar coordinate system.

The direction (D) value obtained from Algorithm-4 with the vertical destination point (either Up or Down), provides the condition for choosing the combination of equations described in Section 3.1. UAV traversing to all the generated NBVs resulted in the completion of the reconstruction process. The simulation result of the

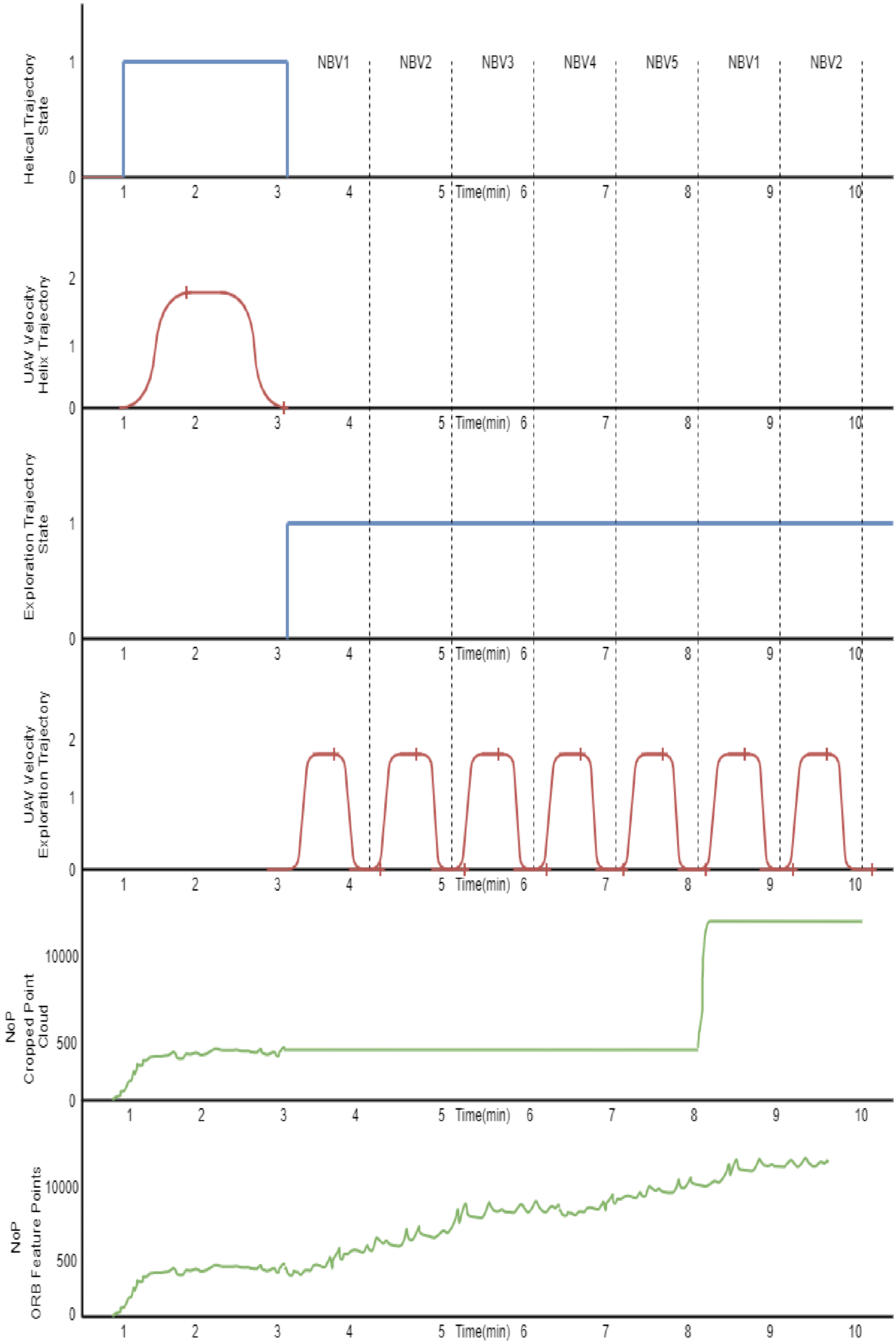


Figure 3.10: Sample illustration of proposed algorithm's parameters over time.

reconstructed cylindrical object of interest using the proposed method is shown in Figure 3.12.

Figure 3.12: The complete 3D Point cloud model reconstruction(right) of a cylindrical object(left) using the proposed method.

3.7 End of Reconstruction

On performing trajectory to all the view-points established by Algorithm 2, point clouds are progressively built.Overall Algorithm shown in Figure 3.13 includes the sub functions that are describes in previous algorithms.Limited power source in the vehicle demands a termination condition to end the reconstruction process. In our algorithm, the total number of feature points are considered to decide the end of reconstruction. If the number of feature points remains within a range for a pre-defined period, then the reconstruction process is said to be completed. Our algorithm relies on these criteria because the generation of NBVs is based on the textures. The textures resulting in weak map points always satisfies the threshold condition, which leads to recurrence of NBVs. Thus we rely on the total number of feature point generated instead of looking for the number of NBVs reaching null.

Overall Algorithm:

Data: Initial ORB feature points from ORB-SLAM2 system, $P = \{p_1, p_2, p_3, \ldots\ldots p_n\}$ where $o_i \in R^3$. Current position of the UAV, $U = \{u_1, u_2, u_3, \ldots\ldots u_n\}$ where $u_i = [x, y, z, \alpha, \beta, \gamma]$, Centroids of filtered Voxels on surface, $C = \{c_1, c_2, c_3, \ldots\ldots c_n\}$ States of UAV used for transitions, S_{UAV} = {Init_done, Helix_done, Inspec_done, Recons_done}. Number of ORB features, Cloud_width. Termination condition, $F_{THRESHOLD}$(500 points). Number of NBVs, N_{nbv}

Result: Completed 3D point cloud model that consists of regional ORB features, $C = \{O_1, O_2, O_3, \ldots\ldots O_n\}$

function Reconstruction(Object)
begin
  get $S_{UAV} \leftarrow$ State_Transition(T); //Algorithm.1
  if ((Cloud_width @t - Cloud_width @ (t-30)] > $F_{THRESHOLD}$) || $N_{nbv} > 1$) **then**
    if S_{UAV} = Helix_done **then**
      C ←Voxelization(P); //Algorithm.2
      NBVs ← Position_Centroid(C); // Algorithm.3
      D← Direction_To(NBV); // Algorithm.4
      Fly_Inspec();
  else
    Fly_InitPos();
  end
end function

Figure 3.13: Overall algorithm.

Chapter 4

Efficiency Improvement

Improving the efficiency implies completing the quality work in better time. To make the reconstruction process more efficient, different techniques are carried out in this thesis work. In an online non-model based 3D reconstruction process where the reconstruction time cannot be estimated before the start of trajectory, it is essential to find the unnecessary workload of UAV and cut them off from the algorithm. Octree-based point cloud segmentation method in the proposed algorithm provides different options to make the reconstruction process more efficient. They are, tuning the voxel resolution to increase centroid spacing, tuning the NoP threshold to avoid inappropriate voxels, manipulating the NBV position and changing the orientation of UAV to maximize the coverage. Each optimization method has its individual and combined impact on the efficiency of the reconstruction process, which will be discussed in the following sections.

4.1 Tuning Voxel Resolution and NoP Threshold

Voxel resolution denotes the size of the cuboid-shaped volumetric space that segments the point cloud model in an octree based point cloud segmentation. The detailed explanation about voxels is presented in Section 2.3. Voxel resolution creates a significant impact on the time of reconstruction. Also, voxel resolution, along with NoP threshold limits influences the reconstruction quality. The following section discusses how the voxel resolution decides the time of reconstruction process and how the combination of both voxel resolution and the NoP threshold affects the quality of 3D point cloud model reconstruction.

Contribution of Voxel Resolution

Contribution of Voxel Resolution Voxel resolution is a user-defined value. The octree data structure segments the point cloud model until the defined voxel resolution is reached. The resolution value can be set in such a way that one voxel or several connected voxel grids completely encloses the point cloud model. To achieve a micro-level inspection, it is desirable to choose the resolution value such that several voxels enclose the point cloud model. In our proposed method, the filtered voxel centroids are used to define the regions that need to be focused on improving the feature points. Moving these centroids apart from each other increases the efficiency of exploration task. It helps to cover the uncovered regions in less time. The distance between the centroids shall be increased by increasing the voxel resolution. An added advantage of increasing the voxel resolution is that the number of centroid points are reduced when the voxel resolution is increased.

For example, consider a cylindrical point cloud model of measurement 4-meter height and 2-meter diameter. A single voxel shall completely enclose the point cloud model and returns one centroid point if the voxel resolution of 4 meters is set. Whereas, if the voxel resolution of 1 meter is set, then 16 voxels are created which returns 16 centroid points. Thus, voxel resolution is inversely proportional to the number of centroids extracted. These centroid points are later translated on to the polar coordinate system to act as the Next Best Viewpoints or the way-points for UAV. To summarise, increasing voxel resolution has two advantages as listed below.

1. Increased voxel resolution reduces the number of centroids which ultimately reduces the number of viewpoints to be explored.
2. It increases the distance between the centroids, which helps to cover more uncovered regions in less time.

The simulation results of different voxel resolution are shown in Figure 4.1. The red coloured cubes in the image are the centroid coordinates of voxels that are filtered with the constraint, NoP threshold. The steps of the simulation experiment are as follows. The UAV travels to a height of 4 meters from its initial position. The generated point cloud model is segmented using the voxel grid of different resolutions. This simulation result shows that as the voxel resolution is increased, the distance between the centroid points increases. Another inference that could be drawn from Figure 4.1 is that the occurrence of centroid points moves farther in each image. This phenomenon is due to the ratio between Voxel resolution and NoP threshold, which is discussed in the following section.

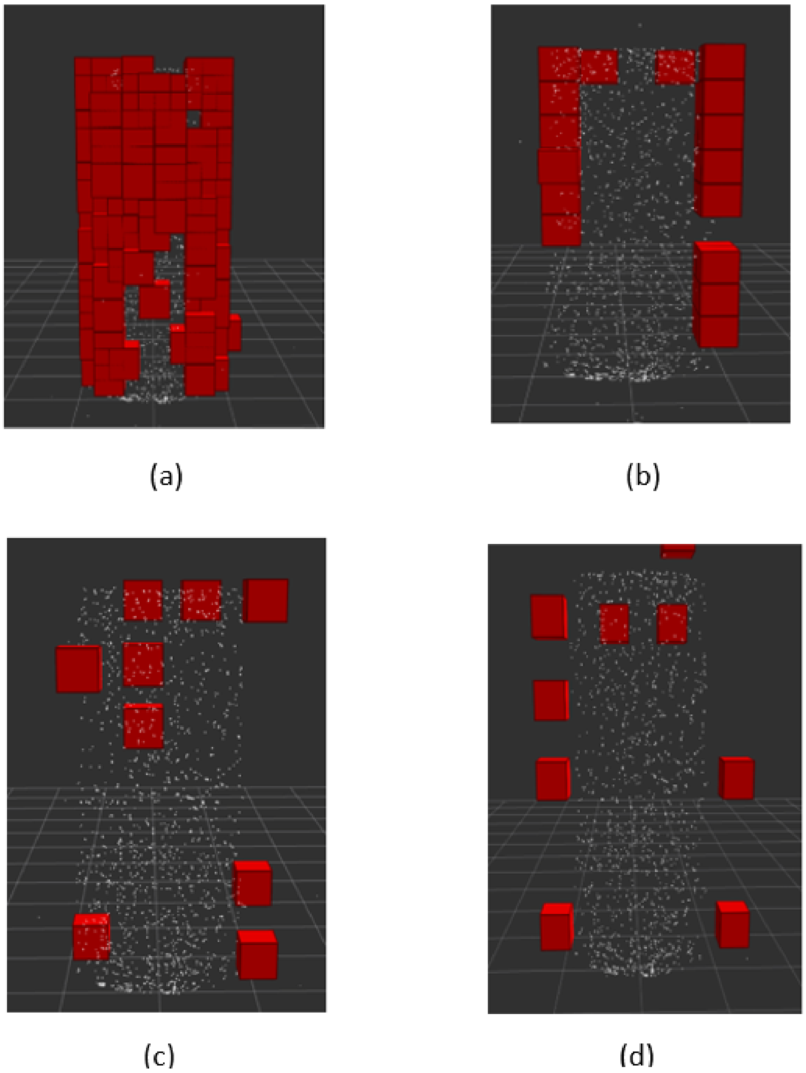


Figure 4.1: Results of increasing the Voxel resolution. Resolution 0.2(a) ,Resolution 0.4(b), Resolution 0.6(c) Resolution 0.8 (d)

Contribution of NoP Threshold

Similar to the voxel resolution, NoP threshold plays a vital role in reducing the reconstruction time. In our proposed method number of points in each voxel is the key piece of information from the octree data structure, which is used to derive the next best view for the UAVs. Hence, refining appropriate NBVs using appropriate NoP threshold limit directly impacts the reconstruction time. The NoP threshold is decided in such a way that the probability of voxel that satisfies the threshold limit is always optimal. The maximum and minimum extremities of probability in filtering the voxels that satisfy the threshold condition results in zero voxels and an abundant number of voxels, respectively. For instance, in the following conditions to filter the voxels, (1) and (2) gives bad results, and (3) gives an optimal result.

1. Threshold limit of one feature point per voxel of resolution 1 meters.

2. Threshold limit of 200 feature points per voxel of resolution 0.2 meters.

3. Threshold limit of 20 feature points per voxel of resolution 0.5 meters.

The first condition will result in zero voxels because the probability of finding a bigger voxel with one feature point is very low. In contrast, the second condition results in a vast number of voxels as in Figure.4.2(a), which ultimately lead the UAV to traverse to ineffective way-points during exploration flight. Hence an optimal NoP threshold limit as in the third condition must be set such that the voxels appear only on the boundaries of the point cloud. Figure 4.2 shows the simulation result of various NoP threshold for the constant voxel resolution (0.4meters). From Figure 4.2, it is evident that as the ratio between voxel resolution and NoP threshold increases, unwanted voxels that are located in the dense point cloud region is reduced.

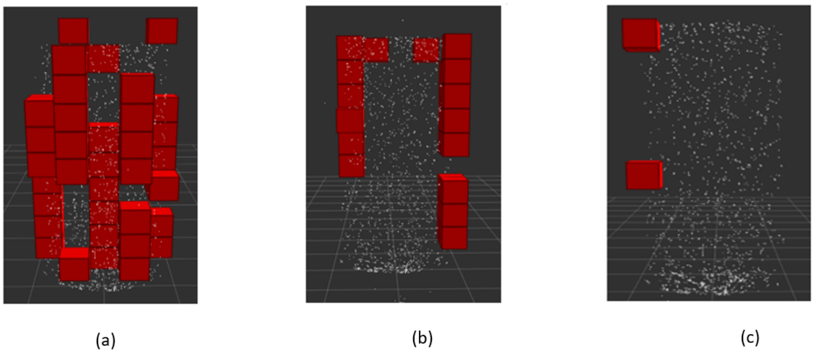


Figure 4.2: Results of varying NoP threshold. NoP threshold 20(a) ,NoP threshold 10(b), NoP threshold 5(c) , constant voxel resolution (0.4)

The contribution of the NoP threshold toward the improvement of reconstruction time is evaluated and presented in Section 5.3.2.

4.2 Manipulating the NBV position & UAV Pitch

The Next best view(NBV) implies the best possible viewpoint for the UAV to generate new surface feature points for reconstruction. In a non-model based exploration task, the best viewpoints are the ones which cover an entirely unexplored region on the surface. To accomplish this, the centroids of the voxels that are translated along the surface normal of the object to the polar coordinate system(as shown in Figure 3.9) are manipulated in such a way that no existing feature points will fall into the FOV of the camera. Hence the available options are

1. Varying the distance from the centre of the object (radius).

2. Moving the point theta degrees around object centre and

3. Translating the point along the vertical axis.

Among these, the first option does not contribute to accomplishing the best view, rather it varies the depth of view and hinders the localization process of ORB-SLAM2 .The second option is taken care of by varying the voxel resolution. As the distance between the centroids increases, the translated centroid points over the polar coordinate system increases its angular distance from its original location on the surface. Translating along the vertical axis, both in the positive and negative direction with respect to the actual viewpoint position, poses a disadvantage. If the viewpoint on the polar coordinate system is translated below the actual position, the top edge of the object is not covered during reconstruction. At the same time, if the view points on polar coordinate system is shifted above its actual position, the FOV may not cover the bottom edges of the object. To compensate for the loss of view, the pitch of the camera is varied along with the repositioning of viewpoints. Pitch is a change in orientation along the lateral axis, as shown in Figure 4.3. Pictorial representation of Pitch, Roll and Yaw. The pitch can be either increased or decreased along the lateral axis. Varying the camera pitch has the following two advantages.

1. If the longitudinal axis (as shown in Figure 4.3) of the UAV is aligned with the surface normal of the object (pitch = zero degrees), then the area covered by the FOV on object's surface increases if the pitch of the UAV is varied.

2. By increasing the pitch, FOV covers maximum usable information while flying at high altitude. Also, the top surface of the object is covered during the reconstruction process. During low altitude flight, the unnecessary ground coverage is avoided by shifting the viewpoints above the actual position as discussed above.

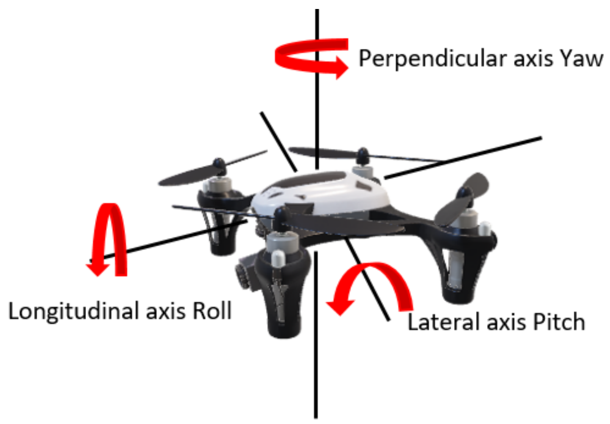


Figure 4.3: Pictorial representation of Pitch, Roll and Yaw.

In our simulation experiment, the translated centroid points on the polar coordinate system are shifted 0.5 meters above the actual position, and the pitch of the UAV is increased by 10 degrees from the initial pitch angle. The pitch configuration of the UAV to optimize the reconstruction process is as below.

$$OLD\,R^U_{pC} = \begin{bmatrix} 0.9082 & 0 & 0.4186 \\ 0 & 1 & 0 \\ -0.4186 & 0 & 0.9082 \end{bmatrix} \; NEW\,R^U_{pC} = \begin{bmatrix} 0.8216 & 0 & 0.5699 \\ 0 & 1 & 0 \\ -0.5699 & 0 & 0.8216 \end{bmatrix}$$

The contribution of changing the NBV position and UAV pitch to the efficiency of the reconstruction process is evaluated in Section 5.3.3.

Chapter 5

Evaluation

The evaluation section of this thesis work is subdivided into four categories. 1) Evaluation of experiment specifications. This part includes the validation of choices made in simulation such as the texture of the object, shape of the object, and radius of the polar coordinate system, 2) Evaluation of the proposed algorithm, 3) Evaluation of efficiency improvement methods that are discussed in Section.4, and 4) Evaluation of termination condition.

5.1 Evaluation of Experiment Specifications

The proposed algorithm for the autonomous 3D surface reconstruction of an object using a UAV is evaluated in a simulated environment. The reason for choosing a simulation instead of real-world UAV is, safety and cost. The gazebo simulation tool along with ROS is used for the experiment. Gazebo is a 3D simulation environment where a realistic complex indoor and outdoor environments could be simulated similar to several gaming physics engines.It consist of various robotic models, wide variety of sensors and convenient programmatic and graphical interface. The specification of the simulation experiment are size, shape, texture and position of the volume of interest,angular velocity of the UAV and size of the outer polar-cylindrical coordinate that encloses the volume of interest. Justification for the choices made on the above specifications is presented in this section.

5.1.1 Surface Texture

Our output is a point cloud model of the surface. Features on the surface are converted to point cloud, hence choosing an appropriate texture is essential in our work. A trial and error method is required to select a suitable texture for simulation. In addition to this reason, ORB-SLAM2 is a feature-based algorithm which ultimately demands features on the surface of the object. As explained before these features are converted to points through triangulation method. Hence texture plays a vital role in building the point cloud as far as cameras are used for perception. LiDAR does not depend on the textures. The intensity of features on the surface makes an impact on the performance of ORB-SLAM2. Because ORB-SLAM2 rely entirely on the map points for tracking, localizing and loop closing. A texture less surface does not trigger the ORB-SLAM2's estimated camera trajectory. Estimated camera trajectory (ECT) is the position of the camera in the Cartesian space estimated by ORB-SLAM2. It calculates this position through point mapping between the left image view and the right image view. It is crucial that ECT travel along with the actual camera trajectory to get an error-free reconstruction. Thus an evaluation of textures is crucial before the start of the experiment.

The texture selection was made with several different samples. The error between the Actual Camera Trajectory (ACT) and the Estimated Camera Trajectory (ECT) is considered for texture selection. If the ECT lags behind ACT, then we can conclude that there are no enough textures present for ORB-SLAM2 to perform point mapping between left view image and right view image. Those textures are rejected for the experimentation. The UAV was made to traverse through the helical path around the cylindrical object-of-interest with different textures such that the stereo camera attached to the UAV always faces the centre of the object. Different textures are used in the experiment, plots describing the trajectory path of ECT and ACT, resultant point cloud model and the error between ECT and ACT for each texture sample used are shown below.

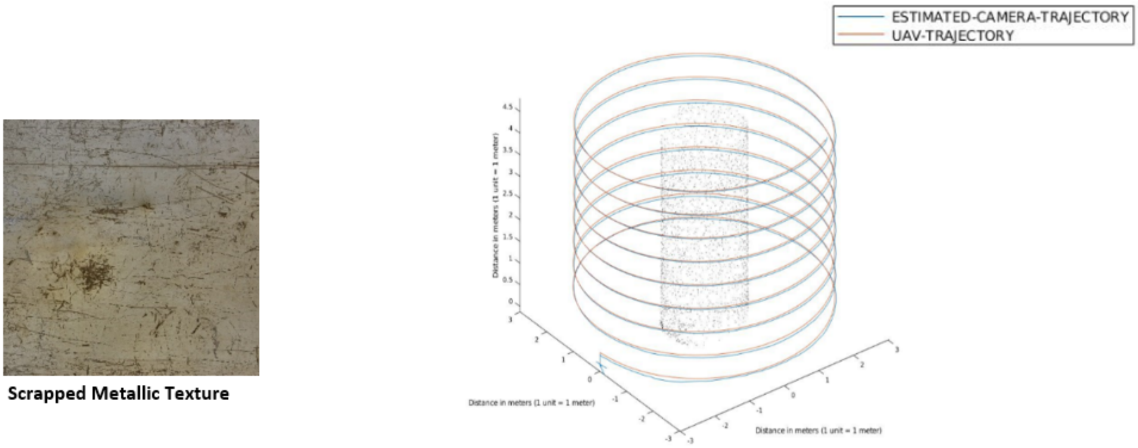


Figure 5.1: Metallic texture where the reconstruction of cylindrical shaped object is complete and the trajectory between ACT and ECT has minimum error.

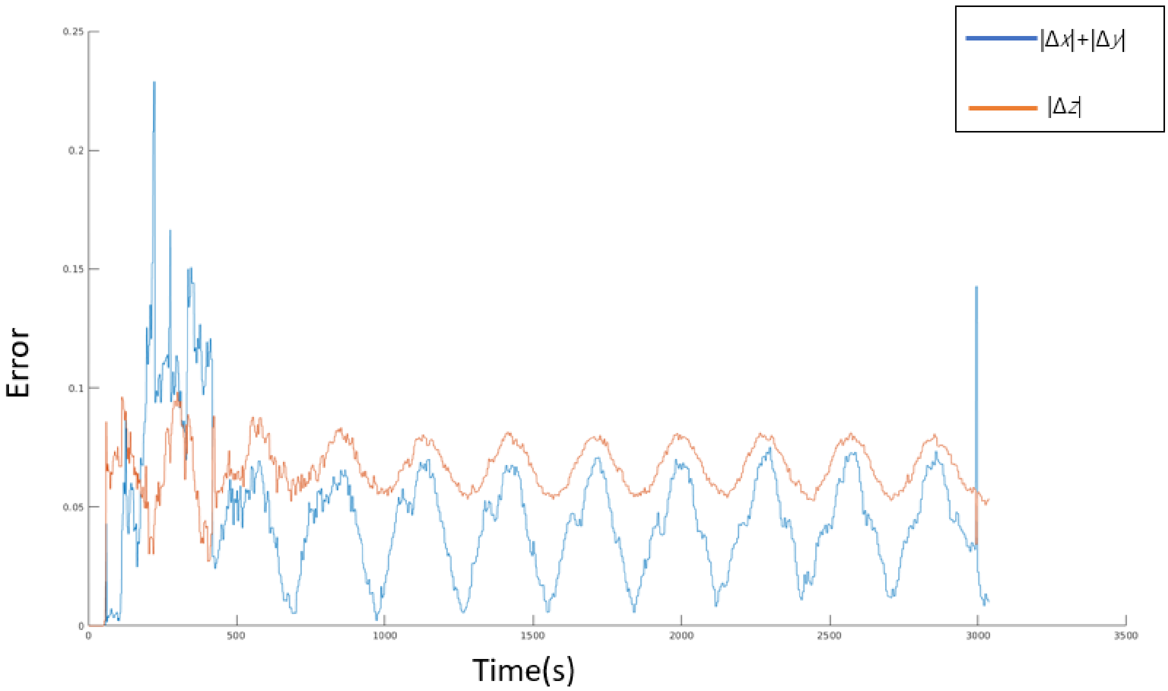


Figure 5.2: Plots of the error $\mid \Delta x \mid + \mid \Delta y \mid$on the x-y plane and the error $\mid \Delta z \mid$ in z-direction between Estimated Camera Trajectory and Actual Camera Trajectory when scrapped metallic texture is used as the texture.

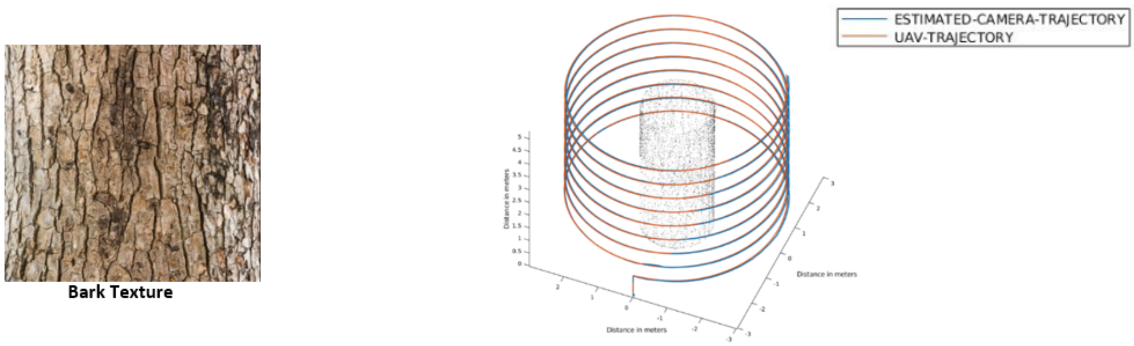


Figure 5.3: A Bark texture where the reconstruction of cylindrical shaped object is complete and the trajectory between ACT and ECT has zero error.

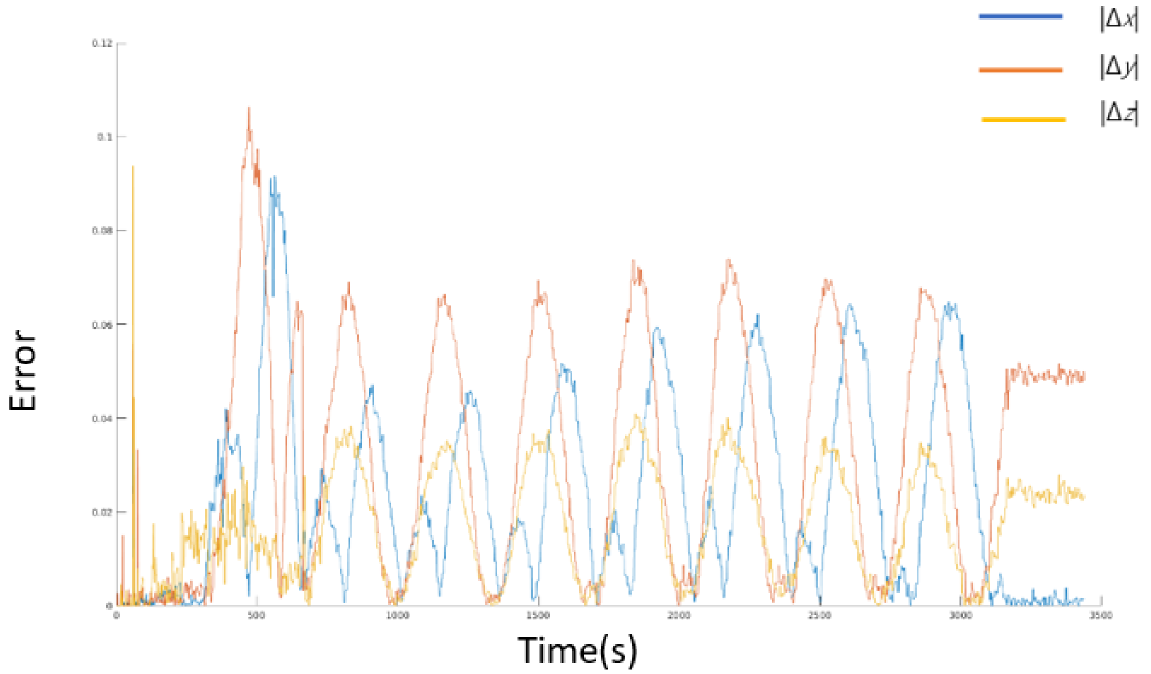


Figure 5.4: Plots of the position error $| \Delta x |$, $| \Delta y |$ and $| \Delta z |$ in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when bark texture is used as the texture.

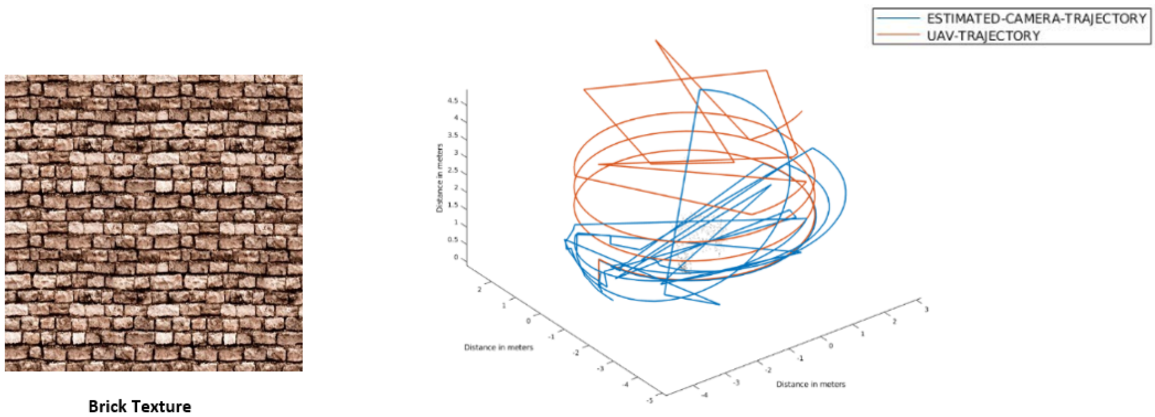


Figure 5.5: A Brick texture where the reconstruction of cylindrical shaped object is not started and the trajectory between ACT and ECT is high and abnormal.

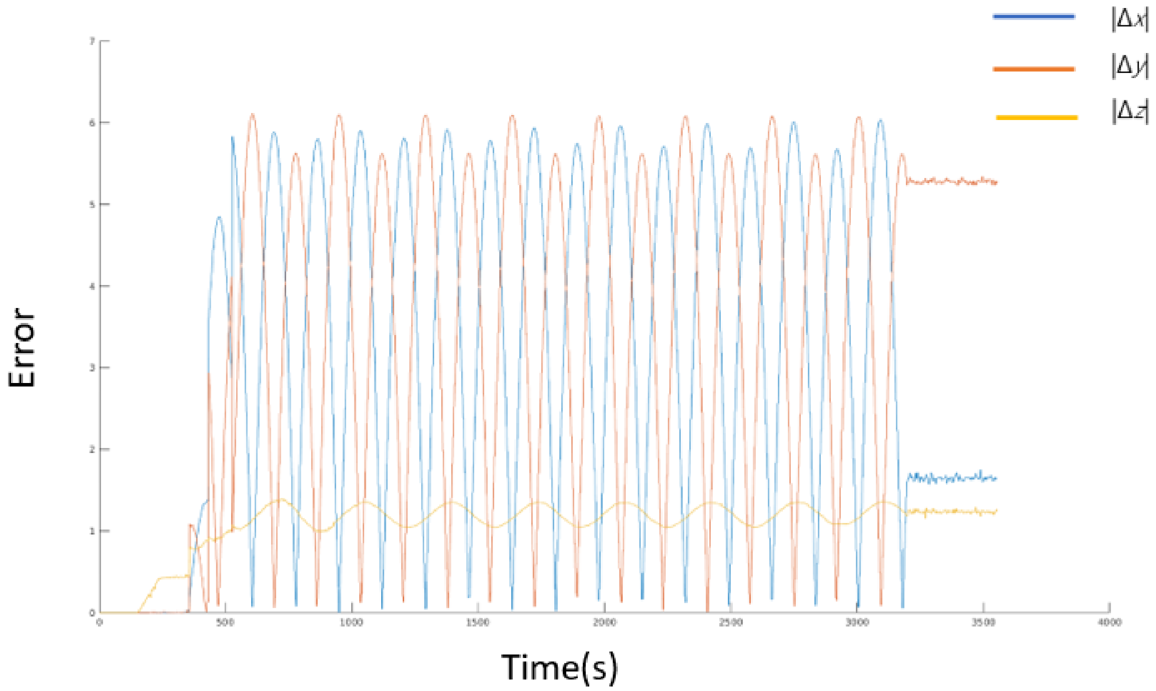


Figure 5.6: Plots of the error $\mid \Delta x \mid$, $\mid \Delta y \mid$ and $\mid \Delta z \mid$ in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when brick texture is used as the texture.

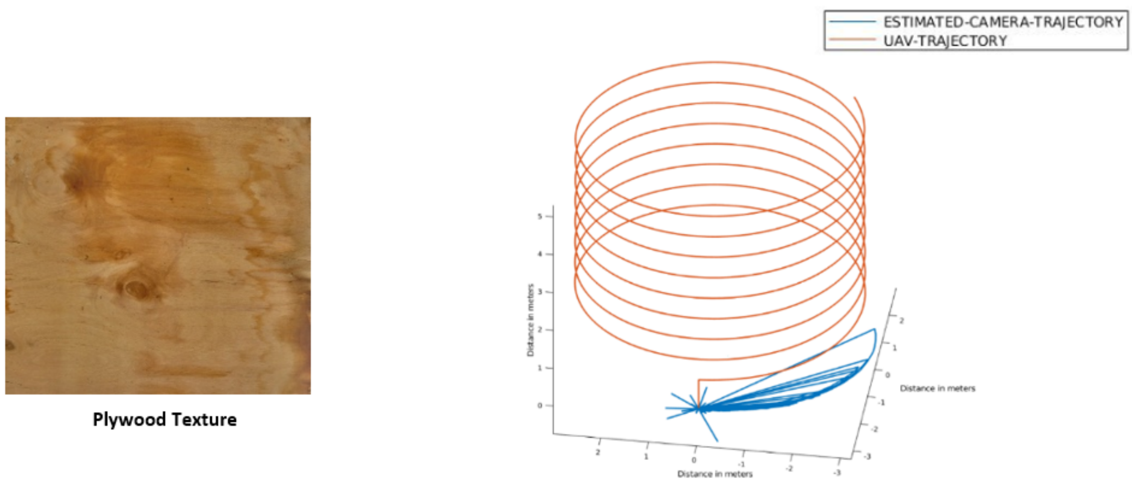


Figure 5.7: A Plywood texture where the reconstruction of cylindrical shaped object is not started and the trajectory between ACT and ECT is high and abnormal.

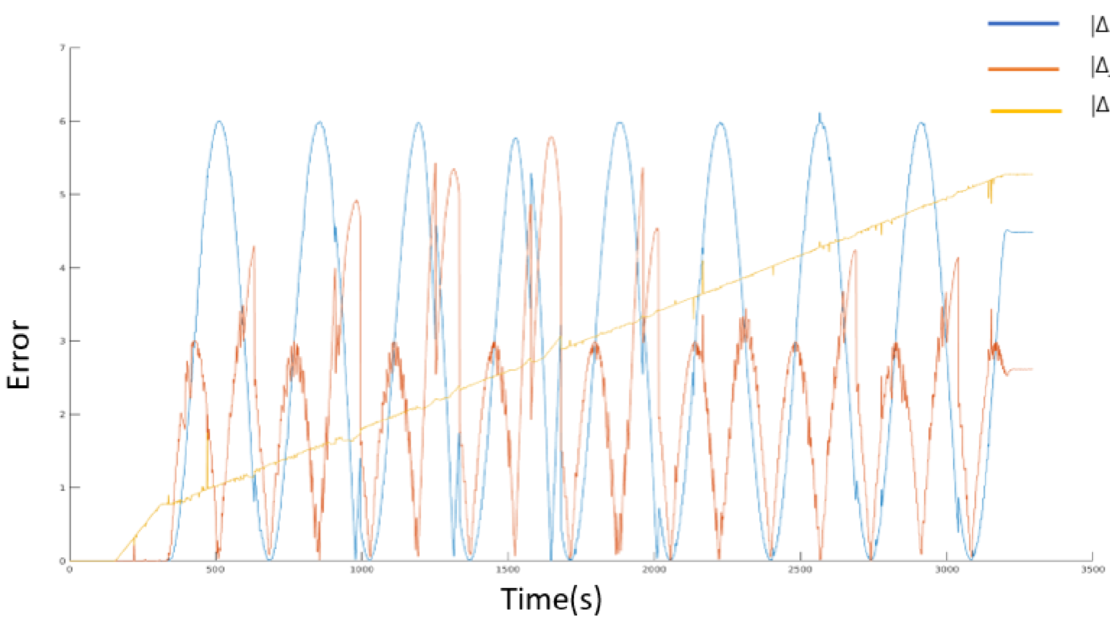


Figure 5.8: Plots of the position error $\mid \Delta x \mid, \mid \Delta y \mid$ and $\mid \Delta z \mid$ in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when plywood texture is used as the texture.

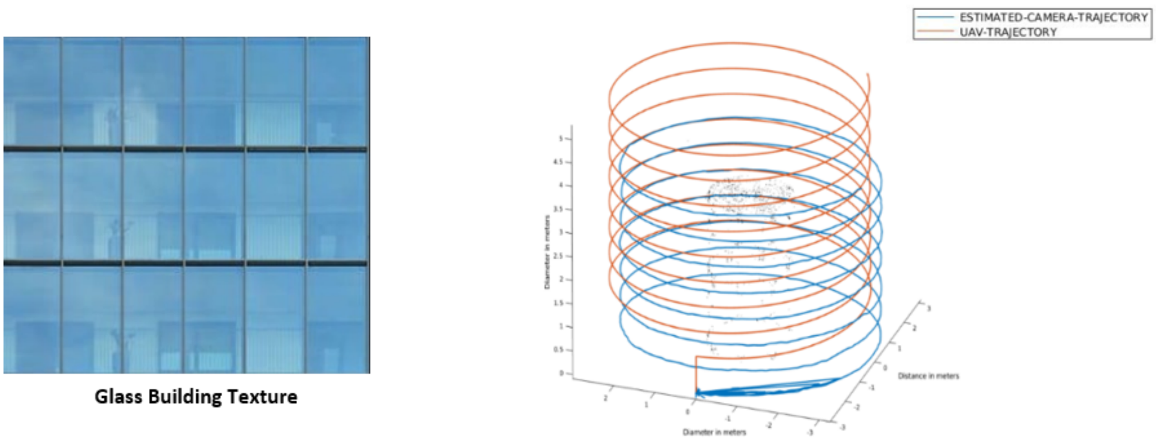


Figure 5.9: A glass building texture where the reconstruction of cylindrical shaped object is Incomplete and the ECT lags behind ACT which produces incorrect reconstruction.

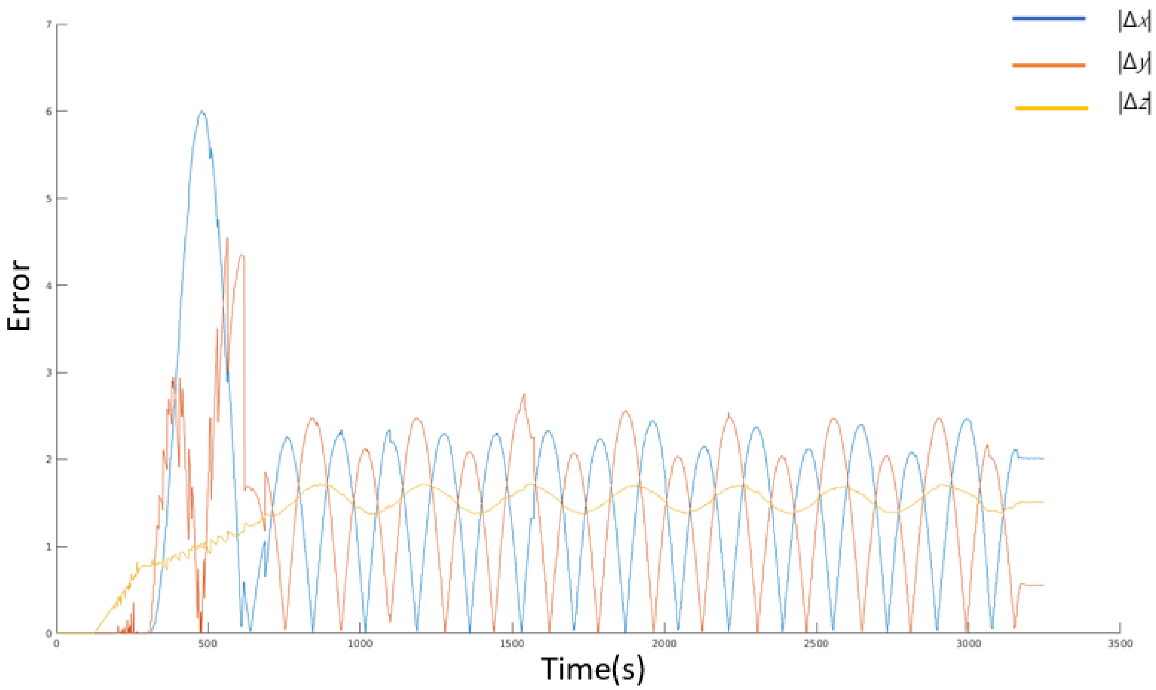


Figure 5.10: Plots of the position error $\mid \Delta x \mid$,$\mid \Delta y \mid$ and $\mid \Delta z \mid$ in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when glass building texture is used.

From the above experiments, it can be inferred that if the estimated camera trajectory matches with the actual camera trajectory due to enough feature points as in Figure 5.2 and Figure 5.4 the point cloud model reconstruction is complete. In Figure 5.6,Figure 5.8 and Figure 5.10 the trajectory graph shows that ECT lags behind ACT or do not follow the expected trajectory due to an insufficient number of features on the surface. Hence, out of the best results obtained using old scratched metallic texture and bark texture, Old metallic texture was used for this thesis work.

5.1.2 Object Shape

Autonomously reconstructing a point cloud model of an object's surface should be versatile; it must perform successful reconstruction of objects with various shapes. It entirely depends on the visual SLAM (VSLAM) the algorithm used in the reconstruction process. We chose to use ORB-SLAM2 algorithm to reconstruct the 3D point cloud model of the surface of a static object in a static environment. Based on the evaluation work done by Patrick (2019) [9], ORB-SLAM2 system is stable and robust of all SLAM algorithm available in a static environment. However, ORB-SLAM2 have limitations with its pose estimation algorithm in dealing with the dynamic feature points. The pose is the spatial position and posture of the camera in the operating environment. Tracking is the process of calculating the change in pose of the robot.

As described by Huaming Qian and Peng Ding in 2019 as described in [25], the pose estimation in ORB-SLAM2 fails if a space point moves with respect to the whole environment. This inference was experienced when the helical trajectory experiment was conducted with a cuboid-shaped object or in other words, objects with sharp edges.

For instance, a cuboid-shaped object enclosed by a polar coordinate system experiences irregularities in the helical trajectory because edges of the cuboid are closer to the UAV path than the points on the surface on all the four faces of the cuboid. The feature points appear to be dynamic(moves closer/away from the camera) to the pose estimation algorithm, which in turn breaks the tracking process of ORB-SLAM2. Thus it is viable to choose a homogeneous surface because the distance between the surface and the camera is always constant, and the feature points remain static to the pose estimation process.

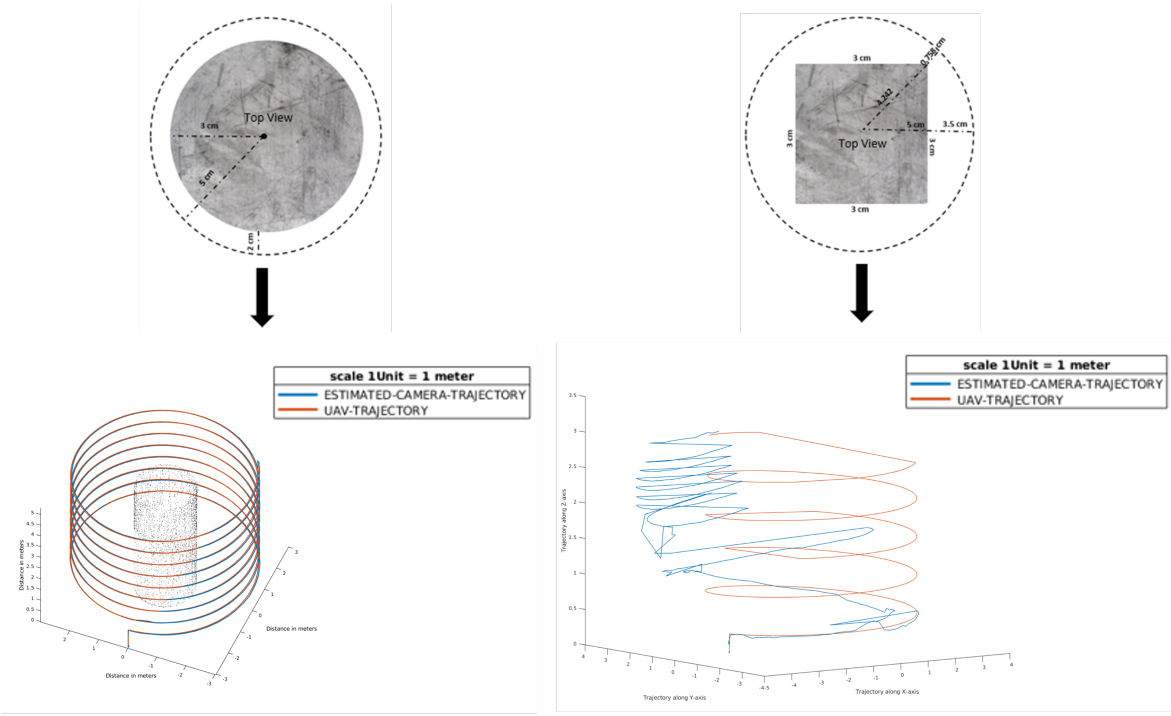


Figure 5.11: A successful reconstruction(left) when a cylinder is chosen and a failed reconstruction(right) using a cuboid object as the object of interest.

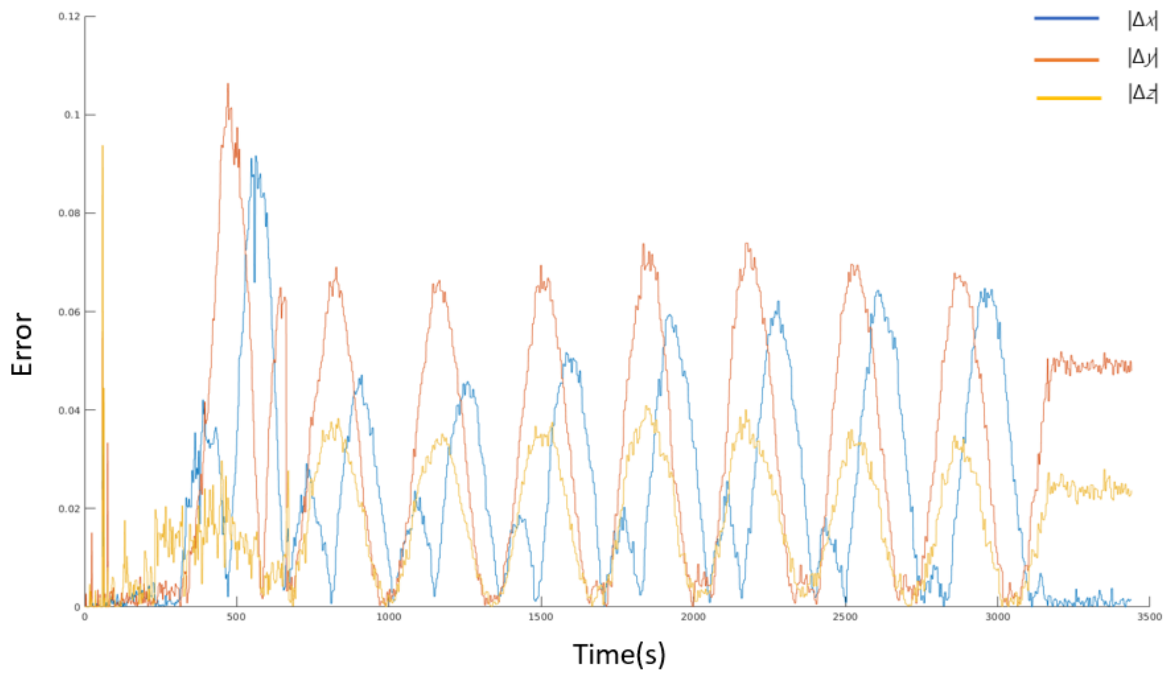


Figure 5.12: Plots of the position error | Δx |,| Δy | and | Δz | in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when object-of-interest with sharp edges are used.

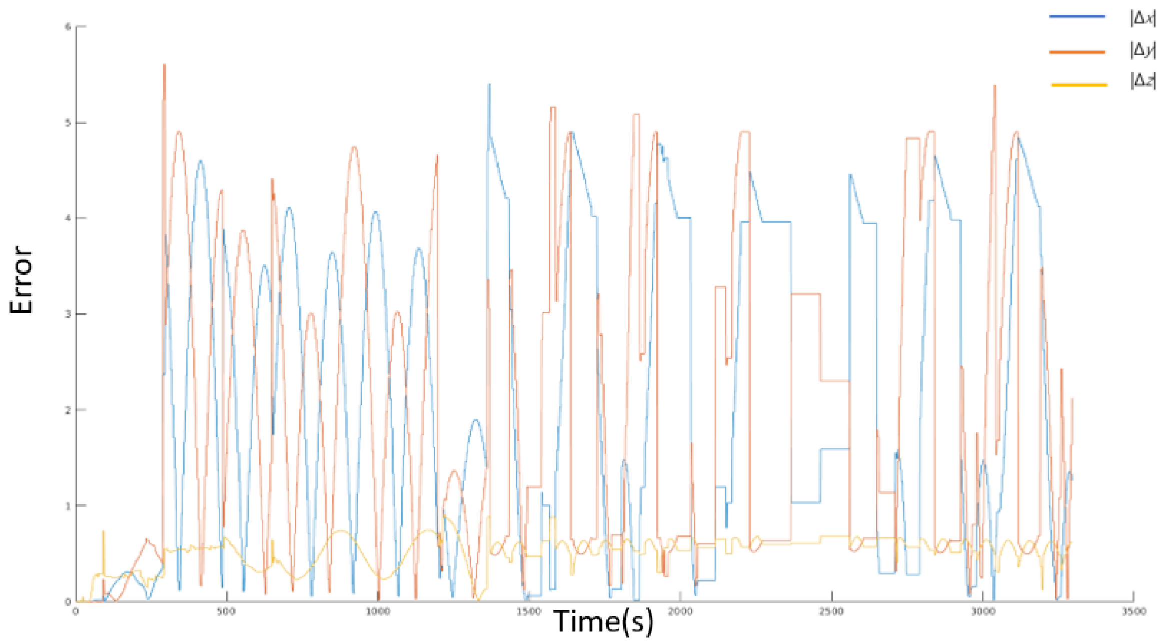


Figure 5.13: Plots of the position error | Δx |,| Δy | and | Δz | in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory, when the object-of-interest is homogeneous.

5.1.3 Polar Coordinate System

Obstacle avoidance is an essential factor to consider in the surface reconstruction process. A UAV during the reconstruction process should not collide with the object-of-interest. To accomplish this, a polar coordinate system is considered around the object as shown in Figure 3.2 such that the object is wholly enclosed and a sufficient distance required for the stereo-vision system to perform pose estimation is provided. Any object or massive structure can be made to fit inside this polar coordinate system. Using a stereo vision camera, the depth of an object can be measured when the object is in the overlapping region of both the left and right view camera. Closer the object lesser will be the overlapping region and vice versa. Hence a minimum distance(radius of the polar coordinate system) away from the surface must be constantly maintained. On executing two different trials with two different radius values, we arrived at a particular distance at which the UAV will perform the smooth surface reconstruction. The experiment is similar to the previous experiment conducted for texture and shape selection. The UAV must complete a helical trajectory around the object without any irregularities, and the error between ACT and ECT should be minimum. Figure 5.14 shows the experimental result conducted with a cylindrical object having old scratched metallic texture on the surface.

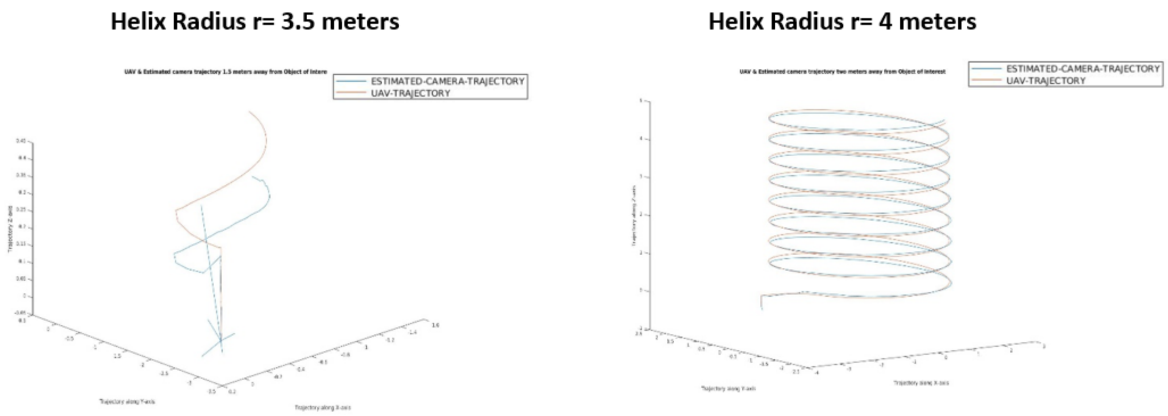


Figure 5.14: Experimental result of UAV revolving around the object at a distance of 1.5 meters away from surface(Left) and 2 meters away from the surface(Right)

The experiment shows that as the distance from the surface increases, the overlapping region between the left image view and right image view increases which facilitates the pose estimation process of ORB-SLAM2. From this inference a conclusion was arrived that the UAV must be made to fly two meters away from the surface of the object to get smooth helical trajectory.

5.2 Evaluation of Proposed Algorithm

Evaluation of the proposed method consists of two parts. One, generating a reference point cloud model which acts as the benchmark solution for the actual evaluation of our proposed method. Two, evaluation of our proposed method, where the results obtained from the helical trajectory and the proposed method are qualitatively and quantitatively analysed.

5.2.1 Creating Benchmark Point Cloud Model

The proposed reconstruction algorithm is a non-model based method. The point cloud model of the volume of interest is not available before the start of the reconstruction process. Thus to evaluate the proposed method, it is inevitable to generate a point cloud model of our own using the trivial solution s(helical trajectory). A helical trajectory around the object of interest results in complete coverage and the required base point cloud model of the object. The base model implies a sufficient number of 3D points in the point cloud model to represent the object's surface. A series of five experiments with the specifications provided in Table 5.1 is executed and the resultant behaviour of the number of point with respect to time is plotted in the graph shown in Figure 5.15.

Experiment Specifications	
Shape of Object	Cylinder
Texture on the surface	Old scratched metal
Height of Object	4.5 m
Diameter of the Object	2 m
Object centred at	Origin
Trajectory shape	Single Upward Helix
Number of rotations in a single helical trajectory	10 rotations per 4.5 m height
Diameter of the polar coordinate system	8 m
Angular velocity of UAV	0.2 rad/s
Vertical velocity	0.2 m/s
Pitch of UAV	0 degrees

Table 5.1: Experiment specification for Single helical trajectory.

At the end of the helical trajectory, an average of 12,826 points are extracted in 708 seconds which constitutes a complete point cloud model of our object of interest. The number of points increases if the number of rotation in the helical trajectory is increased. Because on increasing the number of rotations, multiple coverage of the surface results in the extraction of more number of points. However, on increasing the number of rotations, time of reconstruction process also increases. The benchmark solution obtains a basic point cloud model that represents the complete 3D model of the object of interest. The evaluation of our proposed algorithm is carried out with this reference point cloud model. This point cloud model is used as the source point cloud model in the ICP algorithm.

Experiment	End of Upward Helical Trajectory	
Trial	**Time (s)**	**Number of Points**
1	708	15274
2	708	12688
3	708	12688
4	708	11992
5	708	11492
Average	708	12826

Table 5.2: NoP at the end of the helical trajectory.

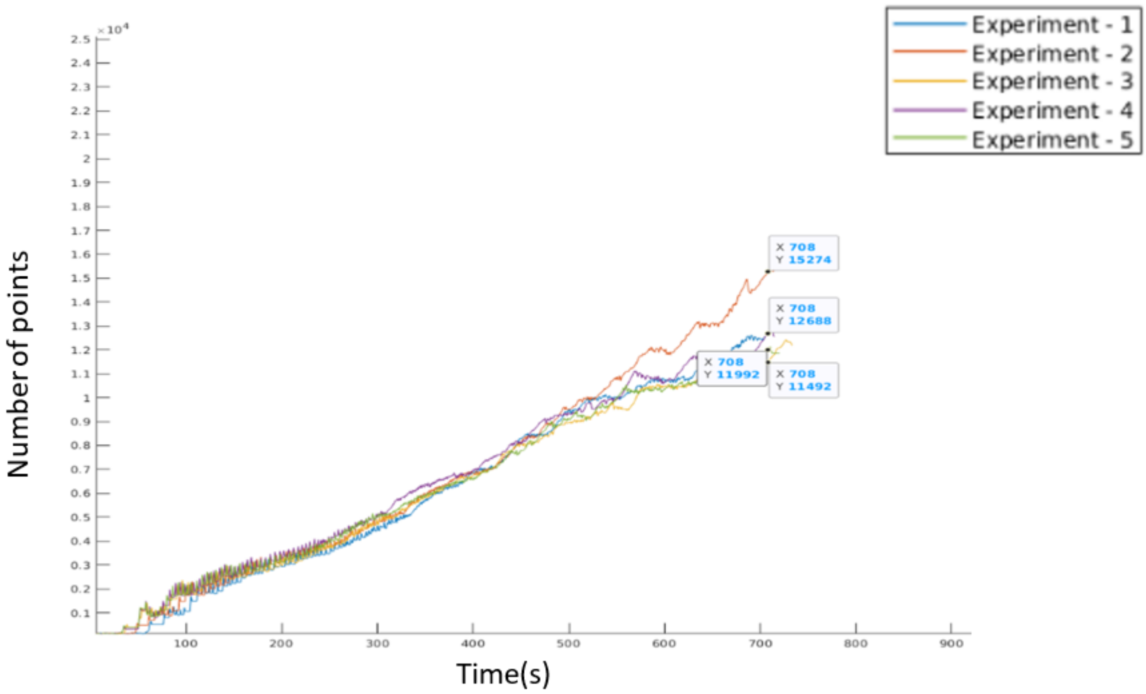


Figure 5.15: behaviour of the number of feature point with respect to time in a helical trajectory.

5.2.2 Evaluation of Proposed Method

The evaluation procedure of our proposed algorithm is carried out in the simulation environment. All the performance improvement techniques discussed in Chapter 4 are introduced in the experiment. The specification of the experiment is shown in Table 5.3. The experiment was conducted five times for authenticity, and the average result is considered for performance analysis.

Experiment Specifications	
Shape of Object	Cylinder
Texture on the surface	Old scratched metal
Height of Object	4.5 m
Diameter of the Object	2 m
Object centred at	Origin
Trajectory shape	Proposed Algorithm
Number of rotations in a single helical trajectory	One rotation per 4.5 m height
Diameter of the polar coordinate system	8 m
Angular velocity of UAV	0.2 rad/s
Vertical velocity	0.2 m/s
Voxel Resolution	(0.4 x 0.4 x 0.4) m
NoP Threshold	$<$ 5 points per voxel
Pitch of UAV	10 degrees

Table 5.3: Experiment specification for evaluating the proposed method.

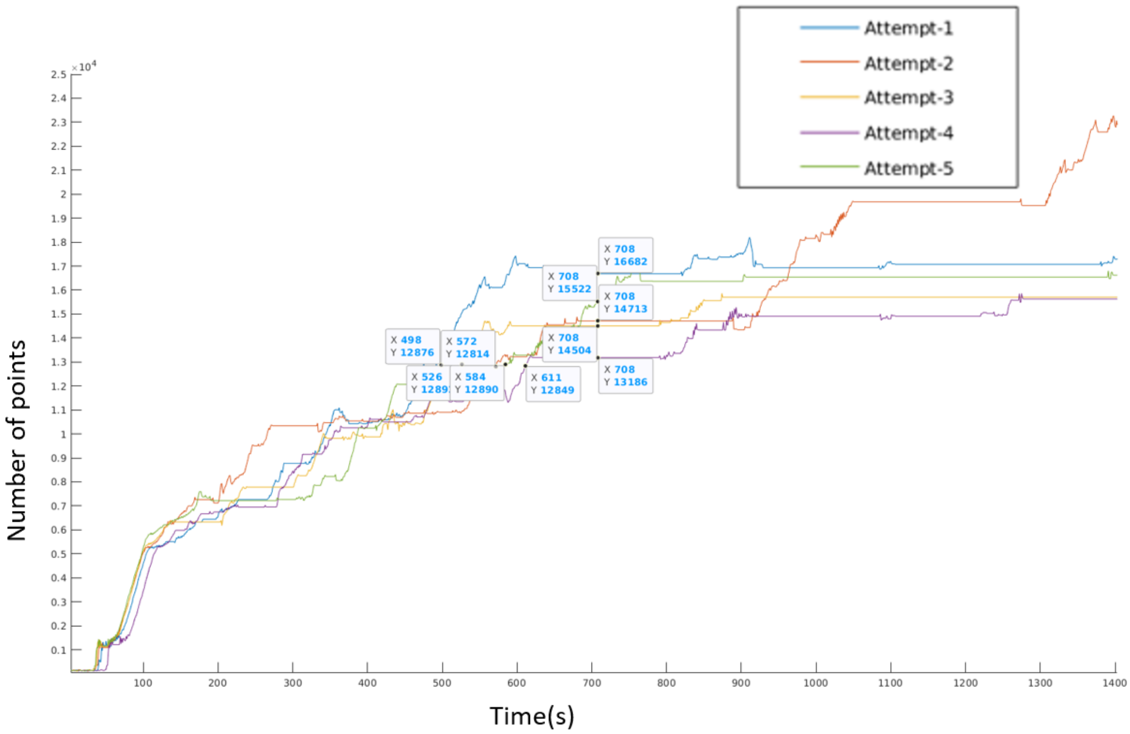


Figure 5.16: Behaviour of number of feature point with respect to time in the proposed method

Experiment Trial	**Proposed Method (Best in Quality)**		**ICP Fitness Score (with Helical Point Cloud)**
	Time (s)	**Number of Points**	
1	498	12876	0.9911
2	526	12892	0.9943
3	572	12814	0.9926
4	584	12890	0.9943
5	611	12849	0.9967
Average	558	12864	0.9938

Table 5.4: Best in Time with benchmark NoP

Experiment Trial	**Proposed Method (Best in Quality)**		**ICP Fitness Score (with Helical Point Cloud)**
	Time (s)	**Number of Points**	
1	708	16682	0.9982
2	708	15522	0.9931
3	708	14713	0.9932
4	708	14504	0.9921
5	708	13186	0.9910
Average	708	14921	0.9935

Table 5.5: Best in NoP at benchmark time

From the experimental results, it is proven that the 3D point cloud model reconstruction process executed using the proposed method gives the best result when compared with the helical trajectory solution. The proposed algorithm reconstructs a complete point cloud model in time 150 seconds faster than the time taken by helical trajectory.The performance in terms of time and NoP are presented below. Each efficiency improvement methods involved in our algorithm is evaluated in the following sections.

Performance in terms of time:

T_{Helix} = 708 seconds to obtain 12864 points

$T_{Algorithm}$ = 558 seconds to obtain 12864 points

$$Performance_{at:NoP=12864} = (\frac{T_{Helix} - T_{Algorithm}}{T_{Helix}}) * 100$$

$$Performance_{at:NoP=12864} = (\frac{708 - 558}{708}) * 100 = 21.18\%$$

Performance in terms of NoP:

NoP_{Helix} = 12826 points at 708 seconds
$NoP_{Algorithm}$ = 14921 points at 708 seconds

$$Performance_{at:t=708} = (\frac{NoP_{Algorithm} - NoP_{Helix}}{NoP_{Algorithm}}) * 100$$

$$Performance_{at:t=708} = (\frac{14921 - 12826}{14921}) * 100 = 14\%$$

the above performance calculation shows that, the proposed method performs 21.18 percentage better in terms of time when compared to helical trajectory method and 14 percentage better in terms of NoP when compared to the helical trajectory method.

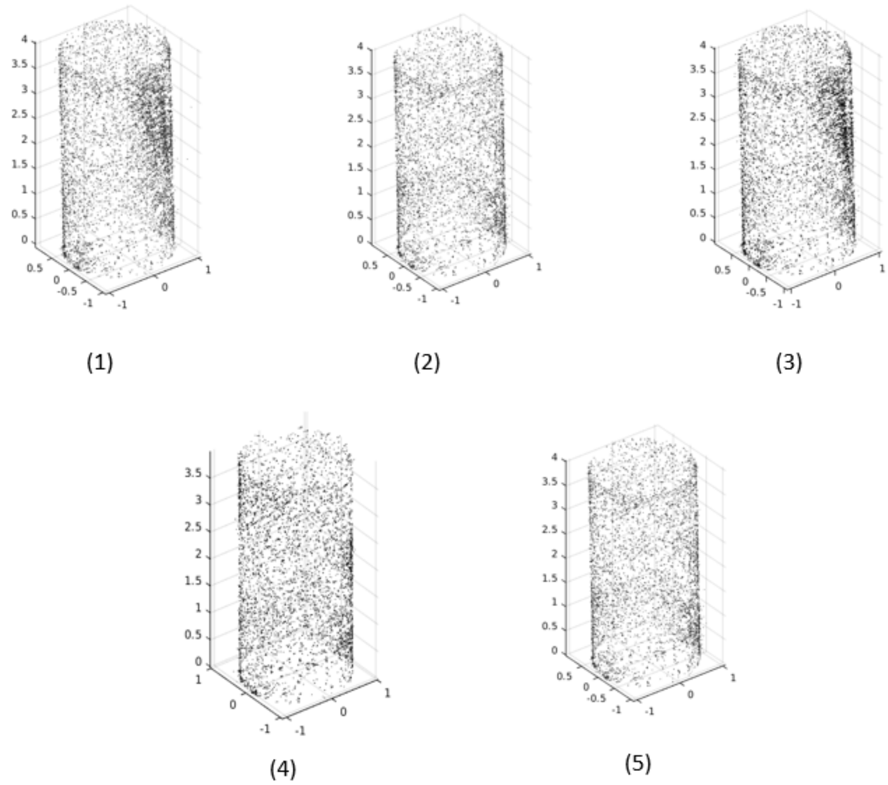


Figure 5.17: The resultant point cloud model obtained in five experiment runs discussed above.

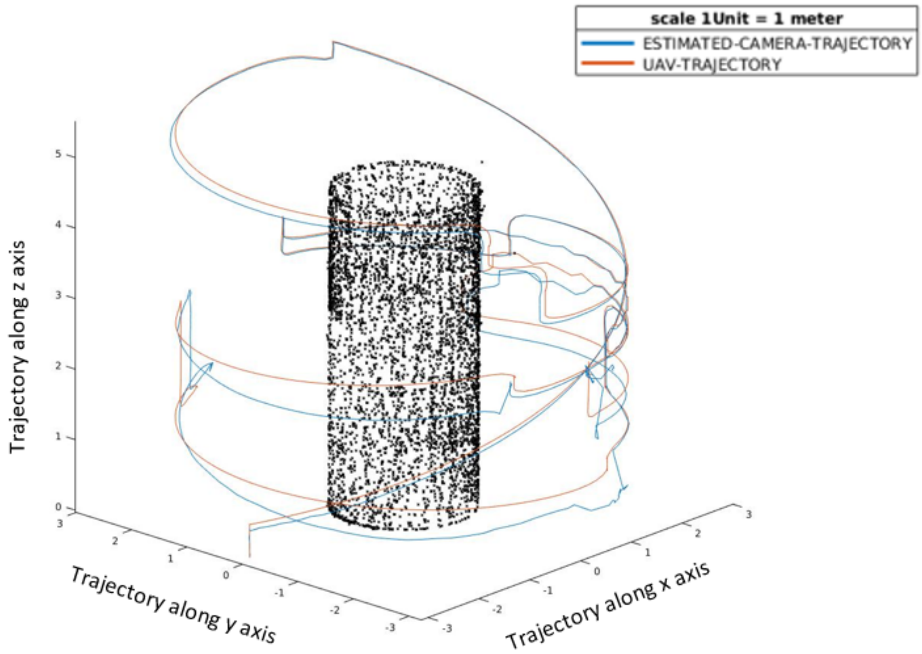


Figure 5.18: The resultant point cloud model with ECT and ACT trajectory using proposed algorithm.

Figure 5.18 presents the point cloud model generated using the proposed algorithm with trajectory made by UAV and estimated camera trajectory. Figure 5.19 shows the error between estimated camera trajectory and actual UAV trajectory along x-y plane(blue) and along z-axis(red). Minimum error between ECT and ACT denotes that the quality of reconstruction is not compromised during the reconstruction process.

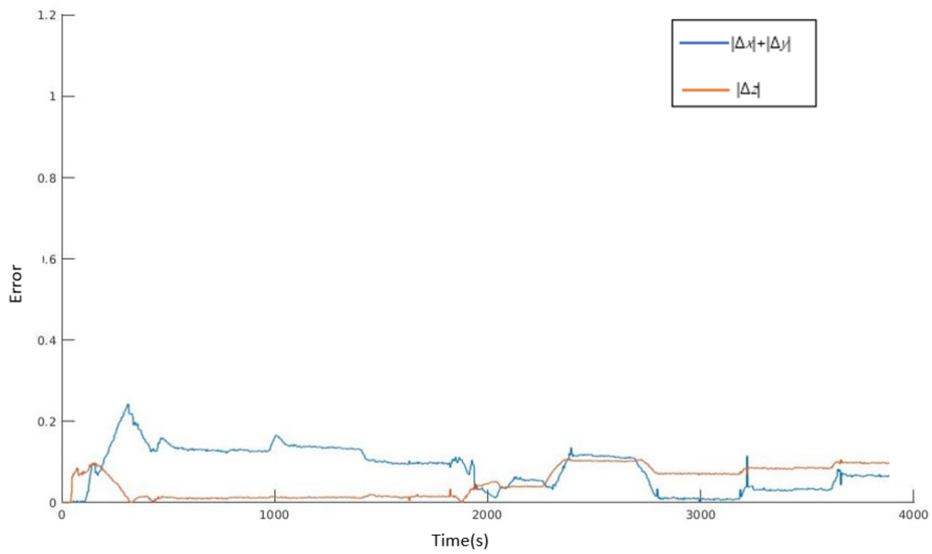


Figure 5.19: Plots of the position error $|\Delta x| + |\Delta y|$ and $|\Delta z|$ in x,y and z-direction between Estimated Camera Trajectory and Actual Camera Trajectory

5.3 Evaluation of Efficiency Improvement Methods

The efficiency improvement methods that are used to reduce the reconstruction time is evaluated using a series of simulation experiments that are specific to each method. Improved efficiency by introducing each methods are individually examined and evaluated. The number of points (NoP) and the time of upward helical trajectory during the evaluation of the helical method is considered as the benchmark solution for evaluation. The test cases that are carried out during the simulation shown in Table 5.6. The results of all the test cases are verified for the completion of the reconstruction process using the ICP algorithm. Finally, the cumulative contribution of all the efficiency improvement techniques are evaluated and discussed.

5.3.1 Evaluation of Voxel Resolution

The first four test cases are executed in the simulation environment with the same object of interest used in our main experiment and double-helical experiment. The simulation results are plotted and shown in Figure 5.20, Figure 5.21 and Figure 5.22. From Figure 5.21, it is evident that the voxel resolution 0.4 gives the best result. It reaches 12,735 points in 1205 seconds. Whether the reconstructed point cloud model with 1205 points represents the expected complete point cloud model of the object of interest is found using the ICP fitness. The point cloud file obtained at 1205 seconds with voxel resolution 0.4 using the proposed algorithm is given as the target file, and the point cloud file obtained using double helix experiment is given as the source file. A fitness score of 0.9461 is obtained using the iterative closest point algorithm. This score implies that there is 94.6% similarities between the given set

Test Title	Test Case	NoP Threshold (points per voxel)	Voxel Resolution (m x m x m)	Manipulating NBV position (metres along z-axis)	Pitch of UAV (degrees)	Angular Velocity of UAV (rad/s)	Vertical Velocity (m/s)
Changing Voxel Resolution	1	15	0.2	0	0	0.2	0.2
	2	15	0.4	0	0	0.2	0.2
	3	15	0.6	0	0	0.2	0.2
	4	15	0.8	0	0	0.2	0.2
Changing NoP Threshold	5	30	0.4	0	0	0.2	0.2
	6	20	0.4	0	0	0.2	0.2
	7	10	0.4	0	0	0.2	0.2
	8	5	0.4	0	0	0.2	0.2
Relocating NBV Position & Varying Pitch of Camera	9	5	0.4	+0.5	+10	0.2	0.2

Table 5.6: the test cases followed for the evaluation of efficiency Improvement methods

of input point clouds at time 1205 seconds. From the above analysis, it is evident that a complete 3D point cloud model of the object of interest is reconstructed using the proposed method in a better time.

Whereas, from Figure 5.22, the voxel resolution 0.6 and 0.8 does not yield a better result when compared to 0.4. The reason for this is that the NoP threshold is maintained constant in all the four test cases. For the given NoP threshold limit the probability of voxels that satisfying the condition is high when the resolution value is 0.4. But, the probability is relatively low when higher resolution values are used. Because finding a bigger voxel that contains less than 15 feature points is a rare scenario in the given texture. Thus for the voxel resolution of 0.6 or above the UAV recursively flies to available NBVs and perform multiple coverage to increase the point count and ultimately increase the number of voxels that satisfies the threshold condition which in turn increases the NBV count. The exploration further continues to the new NBVs. This is the reason why in Figure 5.22 the graph remains stable when the resolution value is set to 0.6 and 0.8. At the same time, the reconstruction time substantially reduces when the resolution is set to 0.4.

Another important point to consider is that, when the voxel resolution is increased, the purpose of localized surface analysis vanishes. Thus voxel resolution must be increased in proportion to the size of the object. If not, the reconstruction process will prolong or may lead to incompletion due to power demand in the UAV.

Test Title	**Test Case**	**Voxel Resolution (m)**	**Time of Flight (s)**	**Number of Points**	**ICP Fitness Score**
Evaluation of change in Voxel Resolution	1	0.2	1520	6884	0.6453
			7731	15521	0.9932
	2	0.4	1520	13112	0.9432
			7731	23297	0.9998
	3	0.6	1520	12509	0.9115
			7731	16218	0.9943
	4	0.8	1520	12509	0.9126
			7731	11720	0.9910

Table 5.7: ICP fitness scores for different voxel resolution at times 1520 & 7731 s

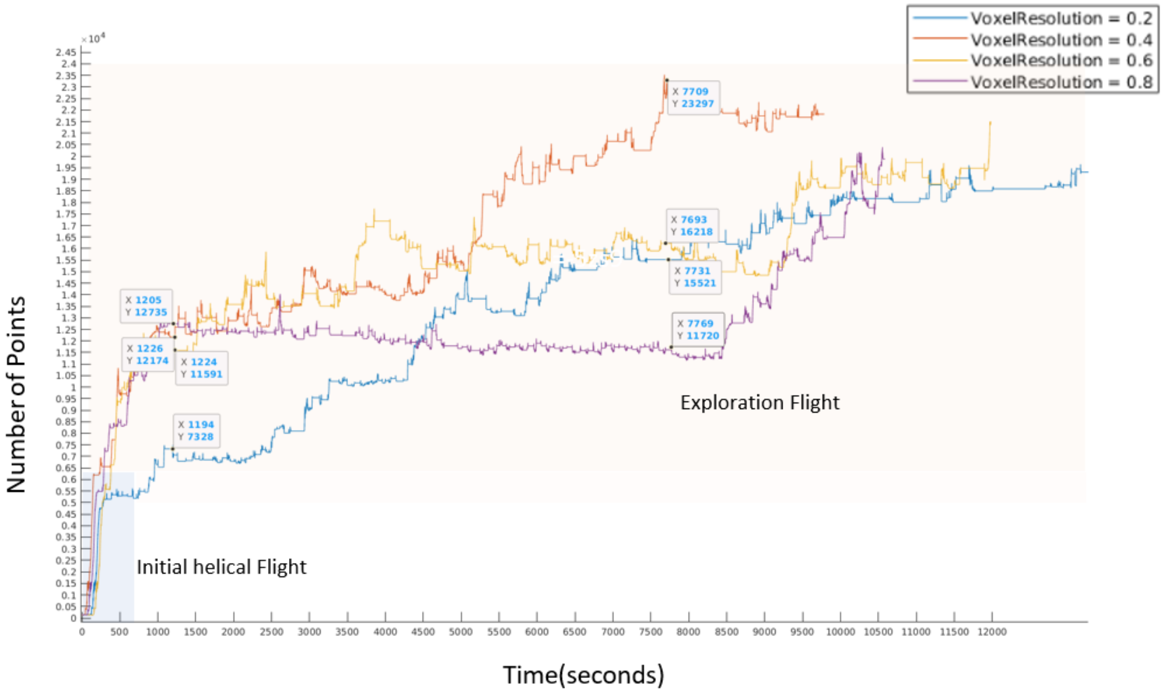


Figure 5.20: Reconstruction behaviour with different Voxel Resolution.

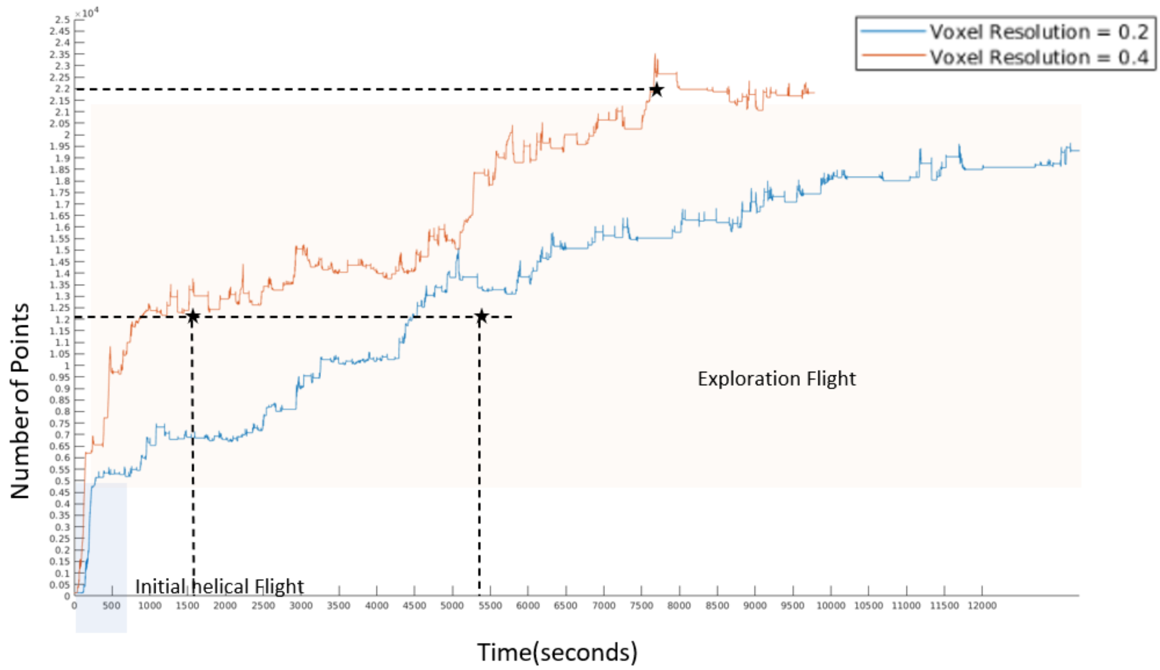


Figure 5.21: Reconstruction behaviour with voxel resolution 0.2 and 0.4

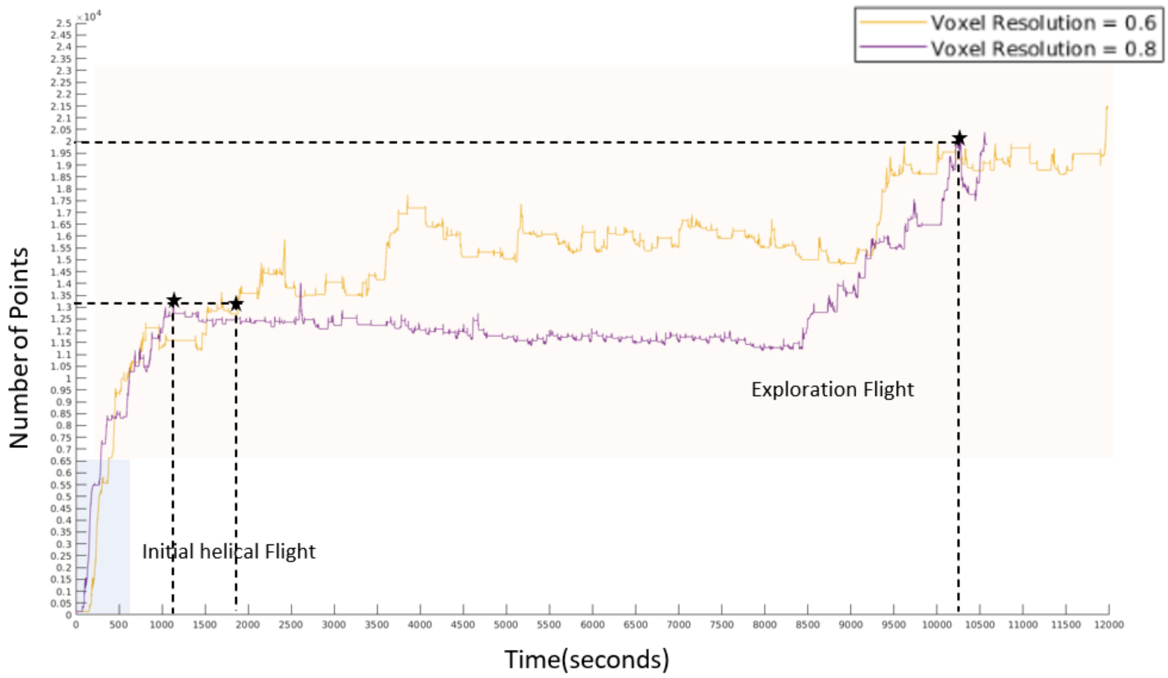


Figure 5.22: Reconstruction behaviour with voxel resolution 0.6 and 0.8

5.3.2 Evaluation of NoP Threshold

The contribution of the NoP threshold in reducing the reconstruction time is evaluated in this section. The test cases five to nine in Table 5.6 are executed in the simulated environment with the same object of interest. The voxel resolution, which showed the best result in the previous evaluation, continues to be the same for this evaluation. Four different threshold values are chosen to analyse the reconstruction performance. The results obtained from each experiment with four different threshold limits are plotted in the graph, as shown in Figure 5.23.

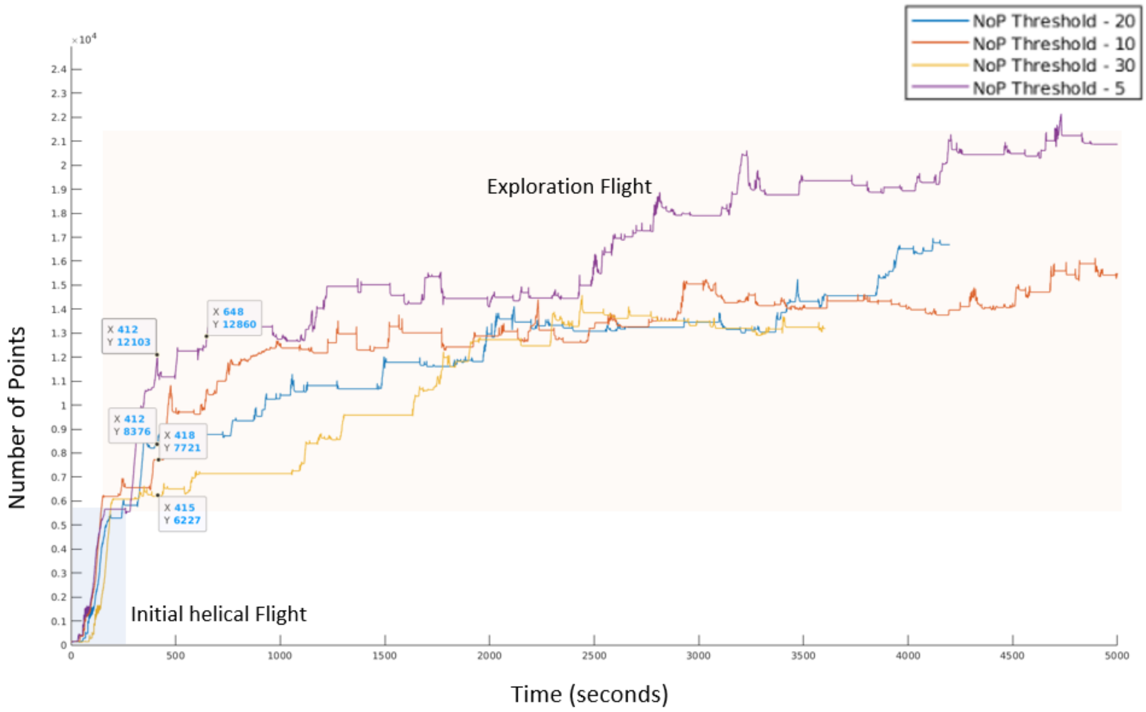


Figure 5.23: Performance variation with different NoP threshold.

The number of points plotted against time in Figure 5.23 shows that the best result is achieved when the threshold limit is set as less than 5 points per voxel. A total of 13,002 points are extracted in 651 seconds. The comparison between the two extremities in the graph shows that the reconstruction performance is 100 per cent higher when the threshold limit is set minimum. As explained in Section 4.1, the unwanted voxels in the dense point cloud regions do not appear when an appropriate NoP value is chosen. Thus letting the UAV make trajectory only to the uncovered regions and reduces the reconstruction time. To verify the completeness of the reconstruction at 651 seconds, the ICP algorithm is made use of to compare the point cloud result obtained at 651 seconds and the point cloud obtained through double helical trajectory. ICP fitness score of 99.81 % is achieved at 651 seconds.

Test Title	Test Case	Voxel Resolution (m)	NoP Threshold Limit	Time of Flight (s)	Number of Points	ICP Fitness Score
Evaluation of change in NoP Threshold	5	0.4	30	651	7144	0.6431
	6	0.4	20	651	8774	0.7743
	7	0.4	10	651	10676	0.8231
	8	0.4	5	651	13002	0.9981

Table 5.8: ICP fitness score for different NoP threshold at time 651 seconds.

5.3.3 Evaluation of Relocating NBVs and Varying Pitch

The ninth test case in Table 5.6 is executed to evaluate the improvement in reconstruction efficiency when NBVs are relocated, and the camera pitch is varied in a positive direction. The experiment specifications, which obtained the best results in the previous optimization methods, remains unchanged in this experiment to get the cumulative performance. The plot in Figure 5.24 shows the results of five experiments executed with the same test case. The result shows that there is a drastic improvement in the reconstruction performance. An average of 13,060 points are extracted in 562 seconds at the end of reconstruction process. And the top surface of the cylindrical object of interest is reconstructed, as shown in Figure 5.25, which was not possible without applying the found efficiency improvement methods.

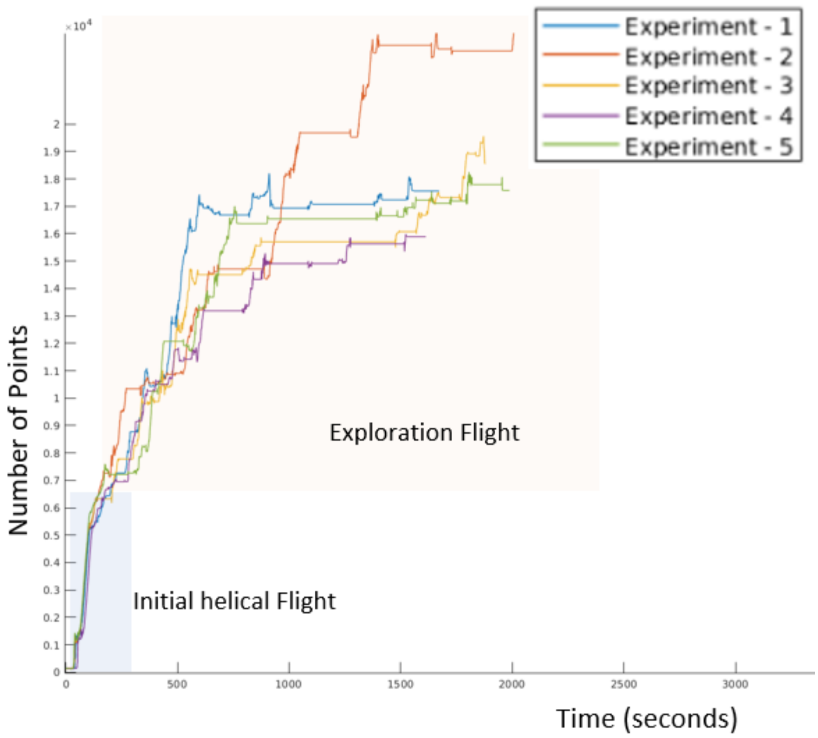
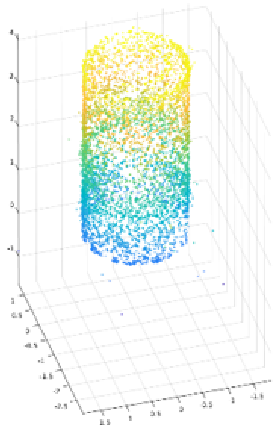


Figure 5.24: Performance improvement by relocating NBVs and changing the Pitch of the camera

Test Title	Test Case	Voxel Resolution (m)	NoP Threshold Limit	Pitch of UAV (degrees)	Time of Flight (s)	Number of Points	ICP Fitness Score
Evaluation of Re-positioning NBVs and Changing the Pitch of UAV	9	0.4	5	10	496	13062	0.9992
	9	0.4	5	10	531	13051	0.9976
	9	0.4	5	10	578	13005	0.9952
	9	0.4	5	10	591	13179	0.9982
	9	0.4	5	10	614	13004	0.9951
Average	9	0.4	5	10	562	13060	0.9970

Table 5.9: Presents the reconstruction trend and ICP fitness score when NBV position is relocated and camera pitch increased using our proposed method.



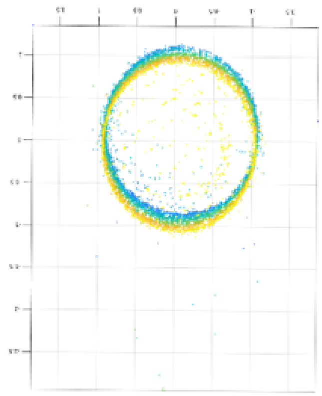


Figure 5.25: 3D point cloud model reconstructed using all the efficiency improvement methods(Left) and top view of the same shows the extracted feature points from the top surface of the cylinder(Right)

The performance improvement by applying each optimization methods is shown in Figure 5.26 . The efficiency of the reconstruction process has increased by applying all the identified optimization methods. Thus the final performance of the proposed 3D point cloud model reconstruction approach has outperformed the trivial solution. Table 5.9 shows that our algorithm has produced an average of 13,060 feature points in 562 seconds, with an ICP fitness score of 99.70%. The performance of the proposed algorithm is calculated in comparison with the helical trajectory experiment presented in Section 5.2.1. The number of points extracted at 562 seconds in the helical method (H_{NoP}) and the proposed method (P_{NoP}) is used to calculate the performance improvement contributed by the proposed method. Table 10 Table 10.shows the NoP obtained in helical trajectory at time 562 seconds. For the chosen object of interest with chosen surface texture, our algorithm has performed 26.89% higher than the helical trajectory. Also, from the ICP score in Table 10 shows that the reconstruction is not complete at 562 seconds of helical trajectory for the chosen object of interest.

Experimental Trial	End of Upward Helical Trajectory		ICP Fitness Score
	Time (s)	Number of Points	
1	562	11086	0.8223
2	562	10516	0.7922
3	562	10151	0.7912
4	562	10238	0.7917
5	562	9473	0.7222
Average	562	10292	0.7919

Table 5.10: NoP obtained in helical trajectory at time 562 seconds.

$$Performance_{at\,562\,s} = (\frac{P_{NoP} - H_{NoP}}{H_{NoP}}) * 100$$

$$Performance_{at\,562\,s} = \frac{13060 - 10292}{10292} * 100 = 26.89\%$$

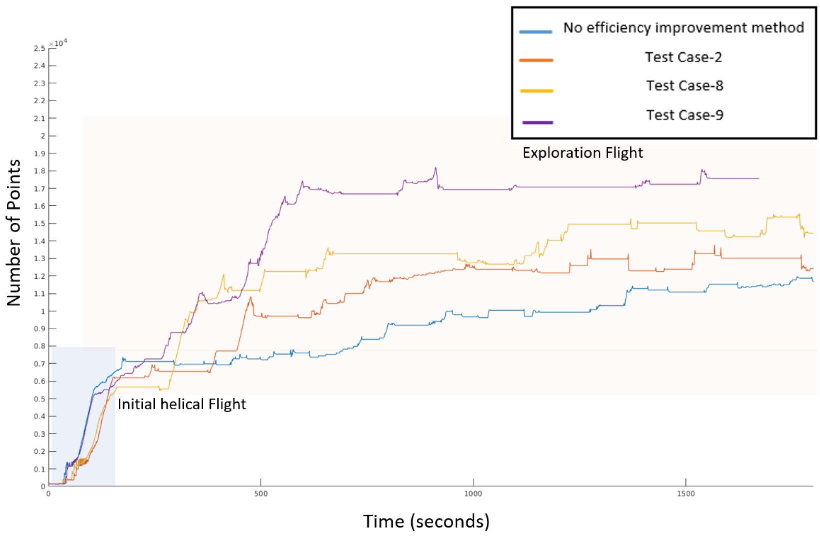


Figure 5.26: Step by step improvement in the reconstruction time using efficiency improvement methods.

5.4 Evaluation of Termination Criteria

In our algorithm the reconstruction termination condition is given based on the saturated number of points or in other words, if the feature point generation is within the pre-defined limit, then the reconstruction process is said to be complete. This termination criterion adapted in our algorithm is evaluated using the double-helical trajectory experiment. The same object of interest (cylinder) with the same texture (old scratched metal) used in our algorithm is subjected to this evaluation process. The UAV is made to execute a double-helical trajectory around the object. The double-helical trajectory consists of upward and downward helix trajectory such that the entire object undergoes multiple coverage during the flight. The number of points generated through ORB-SLAM2 system is recorded and plotted in the graph for analysis. The specification of the experiment is shown in Table 11. Experiment specification for evaluating Helical trajectory and the double-helical trajectory is shown in Figure 5.27. The experiment is repeated six times to build the authenticity factor. The plotted graph is shown in Figure 5.28.

Experiment Specifications	
Shape of Object	Cylinder
Texture on the surface	Old scratched metal
Height of Object	4.5 m
Diameter of the Object	2 m
Object centred at	Origin
Trajectory shape	Double Helix
Number of rotations in a single helical trajectory	36 rotations per 4.5 m height
Diameter of the polar coordinate system	8 m
Angular velocity of UAV	0.2 rad/s
Vertical velocity	0.2 m/s
Pitch of UAV	0 degrees

Table 5.11: Experiment specification for evaluating Helical trajectory.

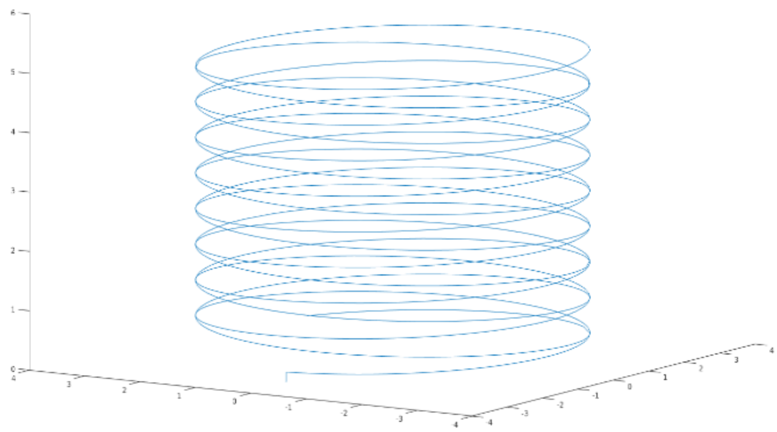


Figure 5.27: Double Helical trajectory

The graph shown in Figure 5.28 reveals that during the upward helical trajectory, the number of feature points increased, which is expected behaviour. During the downward helical trajectory where the FOV covers the same region covered during the upward trajectory, the number of feature point do not drastically increase.

Thus we can conclude that for a given object of interest with the given texture, the number of feature point output from ORB-SLAM2 gets saturated within a range at the end of double helical trajectory. Any point beyond upper helix region shall be considered as a termination criterion to conclude the reconstruction process. Another important inference from this experiment is that there is an improvement in the point cloud quality after the upward helical trajectory. This implies that multiple coverage of parts of the object increase the quality of point cloud. Thus, an upward helix trajectory generated a sufficient amount of feature points to complete the reconstruction process, and the downward helix trajectory improves the quality of point cloud output that was generated during the upward trajectory. Table 12.Show the increasing trend in point cloud density after multiple coverage(Downward helical trajectory)shows the split-up of the number of points at the end of the upward helical trajectory and downward helical trajectory. Among the five experiments conducted, an average of 14,536 feature points are obtained at the end of the upward helical flight in 625 seconds.

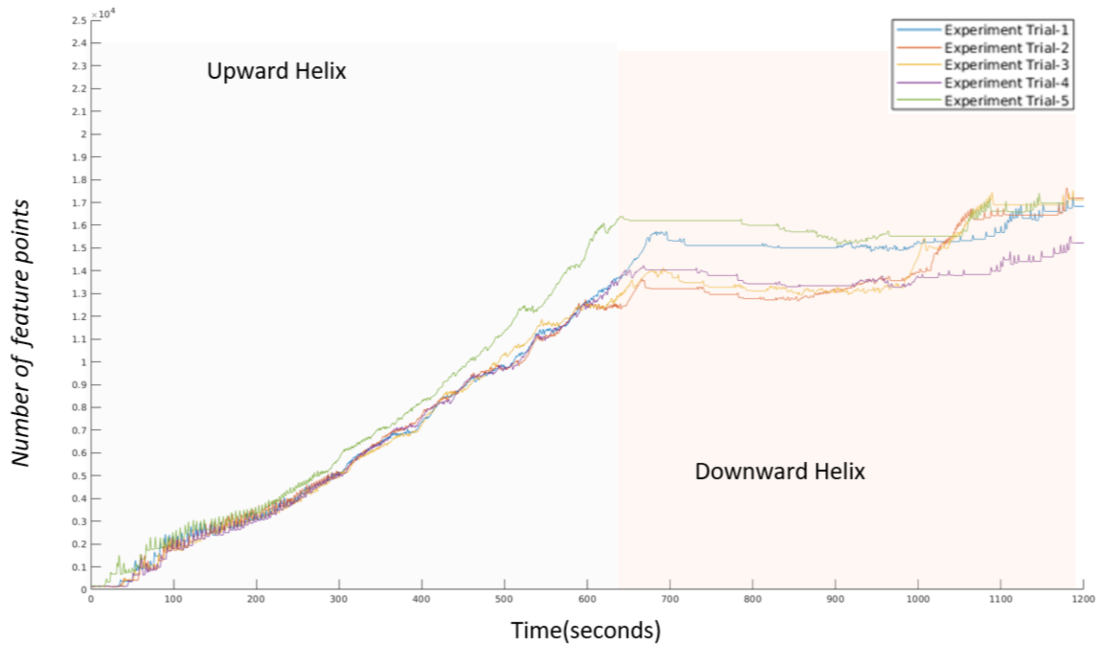


Figure 5.28: Graph plotted with number of feature points against time. Each coloured plot represents an experiment trial. Five trials were attempted during the experiment.

Experiment Trial	End of Upward Helical Trajectory		End of Down ward Helical Trajectory	
	Time (s)	Number of Points	Time (s)	Number of Points
1	625	13899	1155	16607
2	625	13212	1155	16445
3	625	15333	1155	16968
4	625	16202	1155	16607
5	625	14038	1155	14608
Average	625	14536	1155	16247

Table 5.12: Increasing trend in point cloud density after multiple coverage(Downward helical trajectory)

Chapter 6

Conclusion and Future Work

This chapter consists of two sections where we discuss the conclusion of the report and the recommended future work.

6.1 Conclusion

The research questions framed in Section 1.5 is answered with the evaluation results as the proof of concept.Each research questions are answered one after the other.

Q.1 How does the proposed methodology for 3D point cloud model reconstruction perform better in terms of time and quality than the helical trajectory approach?

It is crucial to access that the proposed method for 3D point cloud model reconstruction of an object performs better than the simplest available solution. A straight forward coverage path planning for surface reconstruction is the helical trajectory around the object of interest. An algorithm that outperformed the helical trajectory approach has been developed and evaluated. The evaluation was carried out with the point cloud model obtained using the helical trajectory. The specification of the helical trajectory was framed such that there is an optimal level of overlap in FOV, and no region on the object surface was left uncovered. This complete point cloud model and the time taken to construct was chosen as the benchmark solution for evaluation. This process of generating a benchmark solution is vital because our goal is to reconstruct a 3D point cloud without a reference model. Hence a ground truth point cloud model of the object of interest, which was reconstructed using the helical trajectory is used for the evaluation purpose.

The evaluation result shows that for a chosen volume of interest and chosen texture, our developed algorithm has performed 26.89% better than the helical trajectory in terms of time . This implies that a complete point cloud model was reconstructed using the proposed method took lesser time than the helical trajectory.

Q.2 How does efficiency improvement methods like tuning the voxel resolution, threshold condition, relocating NBVs, and varying UAV's pitch affects the quality of reconstruction?

As the reconstruction process completely relies on several parameters defined in the algorithm. It is vital to analyse how these parameters positively and negatively affect the reconstruction performance. The following parameters are used for analysis in this thesis work, voxel resolution set during spatial segmentation, and the threshold limit to eliminate the ineffective regions (dense point cloud region). In addition to this the physical orientation of the UAV is varied to increase the useful coverage during the flight.

Each methods are individually analysed on how it improves the reconstruction efficiency.Without using any of the methods, the proposed algorithm still complete the reconstruction process in an extended period.But on incorporating the identified methods there is reduction in reconstruction time which was evaluated using nine test cases in Table 5.6. From Table 5.8 (test case 7 and 8) it is evident that a proper tuning of the voxel parameter enhances the reconstruction efficiency. If wrong parameters are chosen, then the reconstruction is incomplete as inferred from Table 5.8 (test case 5 and 6) and Table 5.7 (test case 1).

A most critical analysis from the aforementioned efficiency improvement methods is that in an object-oriented 3D point cloud reconstruction, varying the UAV pitch increases the FOV which ultimately includes a larger area for perception thus improves the efficiency of the reconstruction process. Use of fully actuated UAV have made the tilted flight possible and avoided the use of gimbals, an additional accessory to change the pitch of the camera. In our algorithm where the polar coordinate system is used for UAV trajectory and the camera always facing the object's surface, changing the UAV pitch assist in covering the top surface of the chosen volume of interest which adds credit to the quality of reconstruction. From Figure 5.24, it is evident that the optimization methods have drastically improved the reconstruction time and efficiency of the reconstruction process on the whole.

Q.3 How effective does the termination condition used in our approach determine the end of the reconstruction process?

The primary criteria to terminate the reconstruction process in our proposed algorithm is the saturated number of points. In order to verify if the number of points gets saturated for the chosen object-of-interest, a double helical experiment was conducted. The result of the experiment in Section 5.4 shows that the number of points saturates within a range. However, there is a slight variation in the number of points within this saturated region, as shown in Figure 5.28. This is because the ORB-SLAM2 system regularly updates the extracted feature points where the erroneous points are removed, and new points are added. It is challenging to predefine the upper and lower limit that comprises the saturated region.

In addition to the above drawback, let us consider that a well-defined NoP saturation range is determined to terminate the reconstruction process. In this case, our proposed algorithm achieves the benchmark (helical experiment output) quality of reconstruction in a better time (150 seconds faster) which can be witnessed from Figure 49. If this termination criteria is adopted in our algorithm, then the UAV has to fly for some more time to satisfy this condition which can be inferred from double helical experiment and data shown in Table 5.12. Thus, using the number of feature points to determine the accurate termination time is not the appropriate solution.A work around for this is discussed in the following future work section.

6.2 Future Work

Some of the improvements that could be accommodated in future work are as follows.

1. Some of the NBVs are recursive due to week texture patterns on the surface. As the number of feature points extracted from these week textured regions will not satisfy the NoP threshold condition ever if the generated map points are always less than the threshold value. This is the reason why the total number of points is considered as the termination criteria. At the same time, number of points used to identify the end of reconstruction is not accurate, as the map points saturation takes more time than the completion of reconstruction. If NBV recurrences are avoided in the algorithm, then the return of zero NBVs shall be considered as the termination condition. Also, the reconstruction time will be further reduced.

2. Eliminate the wait time at each NBV position to make the UAV trajectory continuous.In our algorithm this wait time was introduces to stabilize the trajectory. But reducing this wait time to minimum will improve the reconstruction efficiency.

3. Research work on how to accomplish point cloud model reconstruction for different shape of objects. In our proposed method, only objects with homogeneous surface were used as the object of interest. It was identified during the experiment that an object of interest with sharp edges cannot be used for reconstruction.

4. Real-world experiments. By repeating the tests with the same setup should be made possible with the actual hexa-rotor and a modeled object of interest. This real-world experiment will ensure whether the simulation results are appropriate.

Appendix A

Implementation

Implementation of this research work is presented in the appendix section.The tools used, implementation to extract ground truth UAV position and estimated camera trajectory and trajectory of UAV to NBVs.

A.1 Tools and Software

The implementation for this thesis work is based on Robotic Operating System (ROS) [26] along with gazebo robotics simulation tool[(Open Robotics (2019a))]. RotorS Gazebo simulator framework of Furrer et al. (2016) [27], which is used to design our own UAV-models within it as shown in Figure A.1 and develop our own UAV controller.

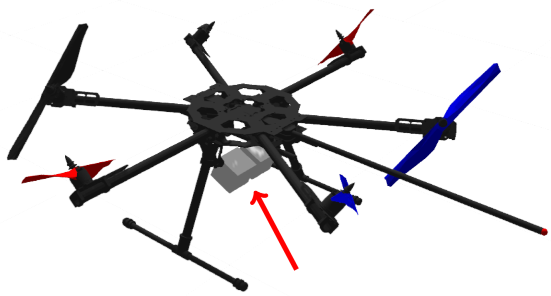


Figure A.1: UAV "betaX" with stereo camera as two cubes below the central base frame (the red arrow is pointing at them)

The controller used in this thesis work developed by Rashad et al. (2019), [28]. It was developed in aim to match a real UAV as close as possible.According to Furrer et al. (2016) [27] the same interface components used in the simulation can be directly applied to real UAV without any modifications. The ORB-SLAM2 system by [2] by Mur-Artal and Tardos (2017) is used for point cloud generation. These point clouds are visualised with rviz [29]. It is used for data visualization for any algorithm.

A.2 Trajectory planning

Two different trajectory planning methods are described in this thesis work. One, initial helical trjectory and two, exploratory trajectory to all the NBV points. The exploratory trajectory which is an online trajectory planning method uses the point cloud provided by ORB-SLAM2.To perform the reconstruction process, the stereo cameras as indicated in A.1 (red arrow) perceives the object surface and sends the stereo frames to ORB-SLAM2 system for further processing. The acquisition of stereo frames are represented as RotorS block in the data flow diagram shown in A.2. The ORB-SLAM2 system extracts the ORB features and sends to the reconstruction component developed for this thesis work.In the reconstruction module the map point outliers are removed using crop box filter and the least point cloud density areas are determined and these coordinates are transformed to polar coordinate system and fed to the graphical user interface(GUI) module. The GUI module receives the new way-point along with yaw, pitch and roll values of the vehicle. The new way-points are determined as explained in 3.3. The yaw angle is set such that the camera always faces the surface of the object.The GUI which provided way-points for the helical trajectory now establishes the waypoints for NBV.The UAV flies to these NBV setpoints by registering the new map points and build the point cloud model.

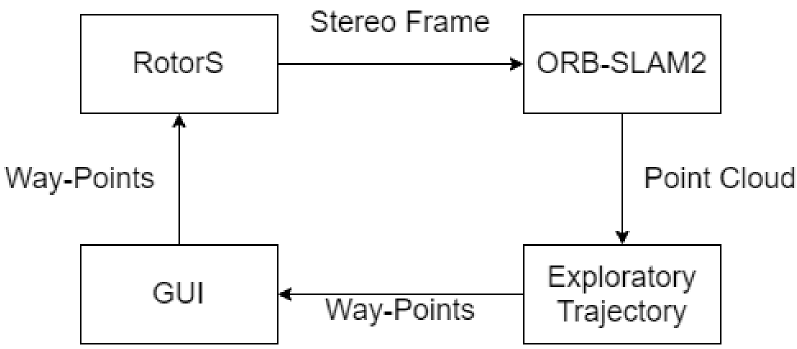


Figure A.2: Data flow diagram that represents the execution of trajectory planning for 3D reconstruction

Bibliography

[1] J. Rowe, *NASA's Centennial Challenges Overview.*, NASA, STMD: Centennial Challenges, 8 2019, https://www.nasa.gov/directorates/spacetech/centennial_challenges/overview.html.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "Hierarchical ray tracing for fast volumetric next-best-view planning," in *2013 International Conference on Computer and Robot Vision*. IEEE, 2013, pp. 181–187.

[4] PCL, *Stanford Bunny Image .*, http://www.programmersought.com/article/5975103939/.

[5] H. Lin, C.-L. Tai, and G.-J. Wang, "A mesh reconstruction algorithm driven by an intrinsic property of a point cloud," *Computer-Aided Design*, vol. 36, pp. 1–9, 01 2004.

[6] A. Zakaria and A. Said, "3d reconstruction of a scene from multiple uncalibrated images using close range photogrammetry," *Proceedings 2010 International Symposium on Information Technology - Visual Informatics, ITSim'10*, vol. 1, pp. 1–5, 06 2010.

[7] S. Hantscher, A. Reisenzahn, and C. G. Diskus, "Application of a surface reconstruction method for material penetrating uwb radar," in *2007 Asia-Pacific Microwave Conference*, 2007, pp. 1–4.

[8] R. Hänsch, C. Wefelscheid, and O. Hellwich, "Three-dimensional building reconstruction using images obtained by unmanned aerial vehicles," vol. XXXVIII-1/C22, 09 2011.

[9] B. Sirmacek, R. Rashad, and P. Radl, "Autonomous uav-based 3d-reconstruction of structures for aerial physical interaction," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 601–605, 2019. [Online].

Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W13/601/2019/

[10] B. Bhanu, S. Lee, C.-C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," in *Proceedings of the eighth international conference on pattern recognition*. IEEE Computer Society Press, 1986, pp. 236–238.

[11] X. Y. Jiang, U. Meier, and H. Bunke, "Fast range image segmentation using high-level segmentation primitives," in *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*. IEEE, 1996, pp. 83–88.

[12] X. Ning, X. Zhang, Y. Wang, and M. Jaeger, "Segmentation of architecture shape information from 3d point cloud," in *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, 2009, pp. 127–132.

[13] J.-E. Deschaud and F. Goulette, "A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing," in *3DPVT*, 2010.

[14] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015.

[15] Z. Dong, B. Yang, P. Hu, and S. Scherer, "An efficient global energy optimization approach for robust 3d plane segmentation of point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 137, pp. 112–133, 2018.

[16] P. Cheng, J. Keller, and V. Kumar, "Time-optimal uav trajectory planning for 3d urban structure coverage," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2750–2757.

[17] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.

[18] C. Connolly, "The determination of next best views," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 432–435.

[19] L. M. Wong, C. Dumont, and M. A. Abidi, "Next best view system in a 3d object modeling task," in *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No. 99EX375)*. IEEE, 1999, pp. 306–311.

[20] P. S. Blaer and P. K. Allen, "Data acquisition and view planning for 3-d modeling tasks," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 417–422.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[22] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[23] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[24] T. Koch, M. Körner, and F. Fraundorfer, "Automatic and semantically-aware 3d uav flight planning for image-based 3d reconstruction," *Remote Sensing*, vol. 11, no. 13, p. 1550, 2019.

[25] H. Qian and P. Ding, "An improved orb-slam2 in dynamic scene with instance segmentation," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. IEEE, 2019, pp. 185–191.

[26] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[27] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *RotorS—A Modular Gazebo MAV Simulator Framework*. Cham: Springer International Publishing, 2016, pp. 595–625. [Online]. Available: https://doi.org/10.1007/978-3-319-26054-9_23

[28] B. Sirmacek, R. Rashad, and P. Radl, "Autonomous uav-based 3d-reconstruction of structures for aerial physical interaction," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 601–605, 2019. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W13/601/2019/

[29] H. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, pp. 1–9, 10 2015.