Cartography M.Sc.

Master thesis

Strategies for a Reduction of Indoor Point Clouds to Purified Room Geometries and their Interactive Presentation

E. Sathish Raymond

Technical University of Munich TUM

TU WIEN TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology

TECHNISCHE UNIVERSITÄT DRESDEN

ITC UNIVERSITY OF TWENTE.

2020

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Environmental Sciences

Institute of Cartography

Master Thesis

Strategies for a Reduction of Indoor Point Clouds to Purified Room Geometries and their Interactive Presentation

submitted by **E. Sathish Raymond**

born on 07.01.1995 in India

submitted for the academic degree

of Master of Science (M.Sc.)

Submission on 10/09/2019

Supervisor Dr.rer.nat. Nikolas Prechtel
Technische Universität Dresden

Reviewer Dr.-Ing. Holger Kumke
Technische Universität München

Cooperation FARO 3D Software GmbH, Dresden

DECLARATION OF AUTHORSHIP

I hereby declare that the submitted master thesis entitled “**Strategies for a Reduction of Indoor Point Clouds to Purified Room Geometries and their Interactive Presentation”** is my own work and that, to the best of my knowledge, it contains no material previously published, or substantially overlapping with material submitted for the award of any other degree at any institution, except where acknowledgement is made in the text.

Dresden, 05th of September 2020 *E. Sathish Raymond*

ABSTRACT

Laser scanning is an advanced technology which is widely used in mapping of indoor spaces. The environment consists of built architecture and secondary features comprises of static internal objects (e.g. a sink or a heater) and dynamic mobile items such as furniture, working materials etc. A clean boundary representation of the architectural shell which has temporal components registered in the scene at data take (e.g. persons or items on a desk) might cause additional disturbances. The semi-automation and automation-based techniques helps to separate secondary features and dynamic objects from other static room geometries (wall, floor, ceiling). The techniques with the use of defined Ransac computer vision algorithm and in-built software algorithms predicts the purified building geometries with high level accuracy and precision on different scale levels. The applied use of semi-automation and automation-based techniques on point clouds has formed an insight and provides a stepping path for transforming 3D geometric information to semantically tagged purified architecture models with an interactive visualization. The purified architectural model and purified point cloud data along with segmented outliers/obstacles are visualized and interacted with higher-end functionalities and visualization elements.

Keywords: LIDAR, Plane detection, Ransac, Automation, Semi-automation, Computer Vision, 3D Modelling, Interactive Visualization

ACKNOWLEDGEMENT

First of all, I would like to thank my supervisor, Dr. Nikolas Prechtel (Technical University of Dresden) and Dr. Oliver Bringmann (FARO 3D Software GmbH), for their continuous assistance and academic supervision support over the period of my thesis. This thesis would not have been possible without Dr. Nikolas and Dr. Oliver’s valuable time and advice in the field of algorithm development, programming, field data support and LIDAR applications. Secondly, I would like to thank Juliane and other co-ordinators for the constant support and help throughout the master study programme.

Special big thanks to all professors and lecturers for supporting and providing me a great knowledge in the field of Cartography. Thirdly, I would like to thank my parents for constant support and help throughout this master degree program and also during the remote master thesis research in the time of pandemic. Finally, I would like to thank all my classmates and its really been a good fun and knowledgeable ride for past two years. I have made a lot of wonderful memories in Germany, which would be life-long inside me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

LIDAR	Light Detection And Ranging
DTM	Digital Terrain Model
DSM	Digital Surface Model
DEM	Digital Elevation Model
BIM	Building Information Model
CV	Computer Vision
AI	Artificial Intelligence
3D	3 Dimension
2D	2 Dimension
CC	Cloud Compare
RANSAC	Random Sample Consensus
PCD	Point Cloud Data
NDT	Normal Distribution Transformation
PCL	Point Cloud Libraries
MDL	Minimum Description Length
GPU	Graphics Processing Unit
LSM	Least Square Model
LRM	Linear Regression Model
LAS Tools	Laser Tools
RGB	Red Green Blue
PLY	Polygon
RQ	Research Questions
VR	Virtual Reality

1 INTRODUCTION

1.1 LIDAR and Indoor Mapping

LIDAR stands for Light Detection And Ranging, which is an active Remote Sensing technique used to examine the object present on the surface of the earth. The basic work principle of LIDAR technology is, by actively firing the laser light/pulse beam of a particular wavelength from the source to the target which actively strikes the target and reflects back to the sensor source and it is recorded. The recorded parameters are the distance between the source and the target, time difference's and the visual properties of the target object. The different wavelength colours of laser pulse's (Green, Blue, Red) are used based on the usage and purpose. LIDAR which is a recent new technology has evolved a lot for past two decades and plays a major role in all fields of science, engineering and technology. In the field of Cartography, it helps to create a very High Definition map which is used in autonomous driving, navigation and mapping. The Cartographic mapping using LIDAR varies based on the platform and acquisition type it operates. The Airborne acquisition using LIDAR helps to create DTM, DSM and DEM of the earth surface whereas the terrestrial acquisition helps to create and construct a 3D object on the earth surface.

Indoor Mapping is a terrestrial based acquisition technique used to map indoors such as rooms, balcony, basement etc of a building using LIDAR technology. The indoors are scanned and recorded as a dense cloud called Point Cloud which is later mapped in 2D and 3D format. Each Point in the Point Cloud has some values and properties such as colour, vertex, normal and co-ordinates present in it. Based on the values and properties one can assign the shape, structure and other visual properties to the object. The Indoor Mapping using LIDAR technology is used in many different fields such as Architecture, Mining, Engineering, Planning, Civil, GIS mapping, etc. The LIDAR technology is one of the easiest and convenient way of converting and transforming the Indoor scanned information to BIM model. The Indoor mapped BIM model can later be rendered, textured, coloured to a real-life model and it can be visualized and interacted in a virtual 3D environment using various tools and technologies.

1.2 Computer Vision with LIDAR

Computer Vision is an interdisciplinary scientific field of Artificial Intelligence that enables the computer to gain high level understanding from algorithms that are developed and enables it to train for understanding and visualizing the real-world entities and objects. The real-world objects that are captured and recorded using LIDAR technology are interpreted, predicted, segmented from noise and visualized using computer vision algorithms. The Ransac algorithm is a robust estimator computer vision algorithm which enables to predict the purified inlier points present in a coplanar Ransac plane from the whole point clouds and eliminate the outliers to maximum probability. The inlier points are visualized and interacted in a virtual 3D environment using various web-based tools and technologies. Computer Vision provides a new path and era in the field of LIDAR Remote Sensing and thus propagates in solving the real-world problems.

1.3 Motivation and Problem Statement

Lidar technology is a recent and an advanced technology which provides a gateway for solving many real-world problems and provides an innovative path of using the recorded geometric 3D information to a semantically tagged 3D models with an interactive visualization. The master thesis is carried out to provide a sematic meaning that the lidar information can be segregated/segmented based on the user needs and purpose. The lidar information can be purified by using both automated and semi-automated based approaches. Due to massive level of point clouds it is difficult to handle without the proper hardware and graphics requirement and thus it will be solved by using down-sampling technique and powerful hardware PC requirements. The master thesis also addresses that the semantic 3D models can be purified based on the user requirement and needs and it can be visualized interactively using web-based tools and technologies.

1.4 Research Questions and Objectives

The main research objective of the master thesis is to reduce the indoor point clouds to a purified geometry with the removal of obstacles and presenting the purified 3D model with an interactive visualization using web-based tools. The purified point clouds are created using both automated and semi-automated techniques and the semi-automated technique which is a program-based approach is then compared to the external software i.e. Cloud Compare. The automated approach is considered for web-based interactive visualization. For attaining the main objective, some of the sub-objectives needs to be carried out and thus are

- Reduction/Segmentation of indoor point clouds using Semi-automation technique
- Reduction/Segmentation of indoor point clouds using Automation technique
- To create a 3D mesh object using Automation technique
- Comparison of finding planes using Python and Cloud Compare
- To create an interactive 3D model using web technology

In order to reach the above-mentioned objectives, some of the following research questions (RQ) needs to be answered and addressed:

- What kind of algorithm and model will be used for plane fitting on 3D points?
- What are the different criteria to make a plane a wall plane?
- What kind of plane points will be extruded in the semi-automated technique?
- What is the effort and certainty in interactively performing the automated task?

To answer all the research questions (RQ), the fore-above objectives needs to be carried out in the master thesis.

1.5 Thesis Structure

The master thesis consists of 9 chapters. The first chapter consists of basic general introduction about the topic such as motivation, problem statement, research questions, objectives that are needed to be carried out for answering the research questions, the general methodology and study area. The second chapter consists of general theoretical background of the project such as literature review of related research studies that are presented, general basic information of the different technical approaches, use of algorithm, use of software and the visualization. The third chapter has complete details about the data collection and preparation. The fourth chapter consists of one of the approaches called Semi-automated technique with detailed description followed by comparative study of the semi-automated technique with external software as the fifth chapter. The sixth chapter consists of second approach called automated technique with detailed process description. The seventh chapter provides content about the complete detailed visualization information of the purified point cloud and purified 3D model. The eight chapter consists of results of the project and discussion of the project results. The ninth chapter provides the conclusion of the whole master thesis project, how well the research questions are explained and answered and the future work of the master thesis in-order to improve the results precision.

The master thesis consists of two parts called Semi-automation and Automation, which are different approaches for creating a purified indoor point clouds and model. The below methodology provides a complete work flow of the master thesis from input point clouds to the final visualization output using different techniques, software's, tools and technologies.

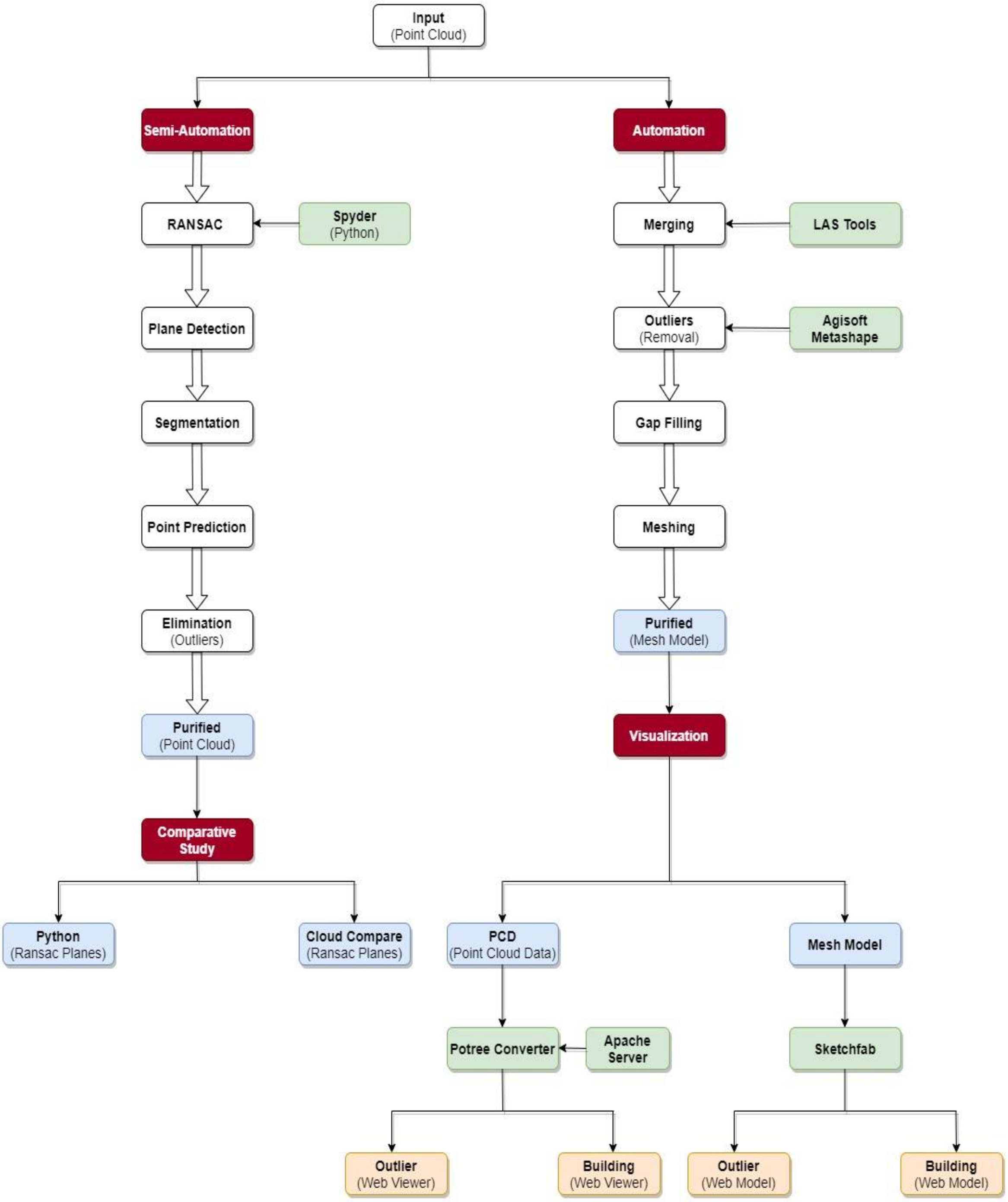


Figure 1: Thesis Workflow (Methodology)

1.6 Study Area

Dresden is the capital city of the German state of Saxony and it's the second most populous city after Leipzig. The city stands on the river Elbe and most of the city's population lives in the Elbe valley. The study area for the master thesis is a three-storey building which is located at South Eastern city of Dresden in Germany and is positioned at the co-ordinates of 51.031259° N and 13.774049° E. The building is operated under a multi-national company named FARO which is known for LIDAR hardware equipment's and software products. For the master thesis, only the 3rd floor of the building is considered.

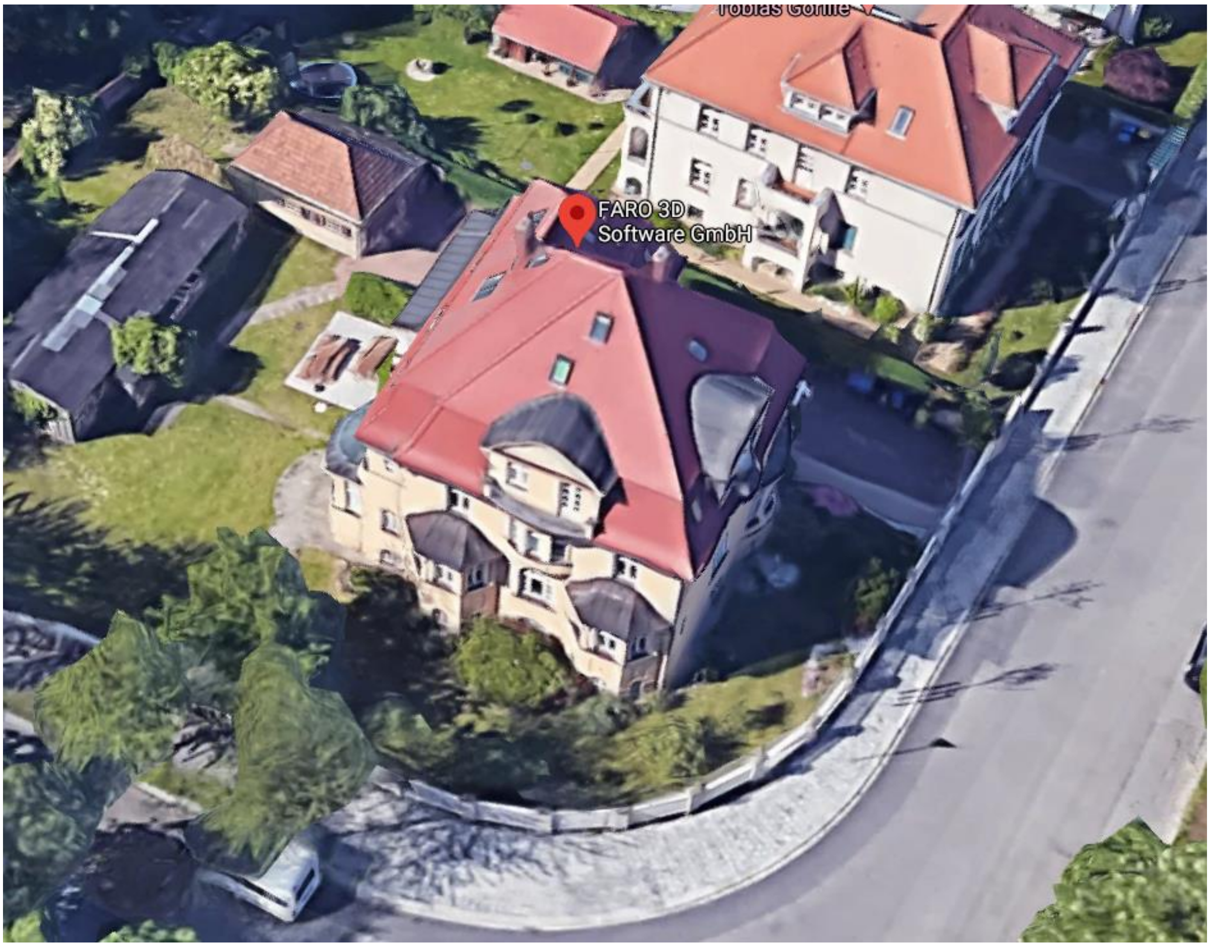


Figure 2: Study Area (FARO 3D Software GmbH) [14]

2 THEORETICAL BACKGROUND

The first sub-chapter of this chapter gives an overview of previous research studies. These studies have described and detected the 3D planes on point clouds and predicted the planar surface object using Ransac computer vision algorithm. The second sub-chapter describes the basic theoretical information of the semi-automation technique, algorithm involved and python packages used for this technique. The third sub-chapter describes the basic comparative study information of the semi-automation technique with external software. The fourth sub-chapter explains the basic theoretical background information of the automation technique and software's involved for completing the process goal. The fifth sub-chapter describes the basic information on the visualization part and the web-tools involved for visualizing the output in the virtual 3D environment.

2.1 Literature Review

The various research works have been studying the Ransac algorithm and trying to predict the planes of indoor environment on 3D Point clouds. The Ransac algorithm for 3D point cloud plane segmentation based on Normal Distribution Transformation (NDT) Cells has been discussed in [1] by Lin Li et al. They use an improved Ransac algorithm for detecting the planes with NDT cell-based approach for neglecting the spurious planes and to remove the noise and outliers on point clouds. The statistical analysis has been carried out for each cell to model the observed points for the set of NDT cells of the point clouds. It has been concluded that the use of Ransac with NDT cell-based method gives high correctness and completeness for predicting the indoor plane environment. In [2] the automated building plane detection using Ransac algorithm is discussed in this literature by D.L. Bool et al. The classified LIDAR data is used as base product for detecting the planes using Ransac algorithm. They have defined the parameters of the Ransac. The roof tops of the buildings are predicted based on the parameters and number of iteration operations. From the literature, it has been concluded that the building roof planes are detected with pre-defined parameters of Ransac and the number of iterations it operates.

In [3] Plane detection and segmentation of Kinect point clouds using Ransac method is discussed in this literature by Kurban R et al. They have used different approach for creating the point cloud. The use of Microsoft Azure Kinect sensor and Kinect camera in it helps to capture, record and store data. The Point Cloud Libraries (PCL) helps them to create a purified point cloud. They have defined a two important parameters of Ransac for Plane segmentation.

From the literature, it has been concluded that the point clouds can be created by only motion sensors and cameras. The parameters in Ransac play an important role for Plane segmentation. In [4] Plane Detection in Point Cloud is discussed in this literature by Yang M. Y. et al. They use Ransac algorithm with Minimal Description Length for detecting and eliminating the wrong planes on point clouds. The literature concludes and provides an idea that the complex geometry on point clouds can be detected with the combined use of Ransac and other approaches, hypothesis and algorithms (MDL).

In [5] a fast-Planar detection method in LIDAR point clouds using GPU-based Ransac method is discussed in this literature by Lan J. et al. The Lidar point cloud data is used as a base reference input and the planes are predicted using Ransac algorithm with defined parameters. They use the new technique such that the predicted planes are converted to mesh by considering only four vertices of a point cloud. The memory usage and speed are controlled and preserved. The literature provides an innovative idea that the man-made infrastructures consists of planar surface and with the combined use of computer vision algorithm and hardware's the planes can be predicted and segmented. In [6] fast and accurate ground plane detection for visually impaired person is discussed in this literature by Zeineldin R.A. et al. They use Ransac algorithm for ground and obstacle detection by predicting and segmenting planes based on the defined parameters and usage purposes. The literature provides an idea that the Ransac algorithm can be used for ground plane and object detection that helps to solve the real-world problems say for navigation of impaired persons.

From the above all mentioned research studies, the following conclusions can be made:

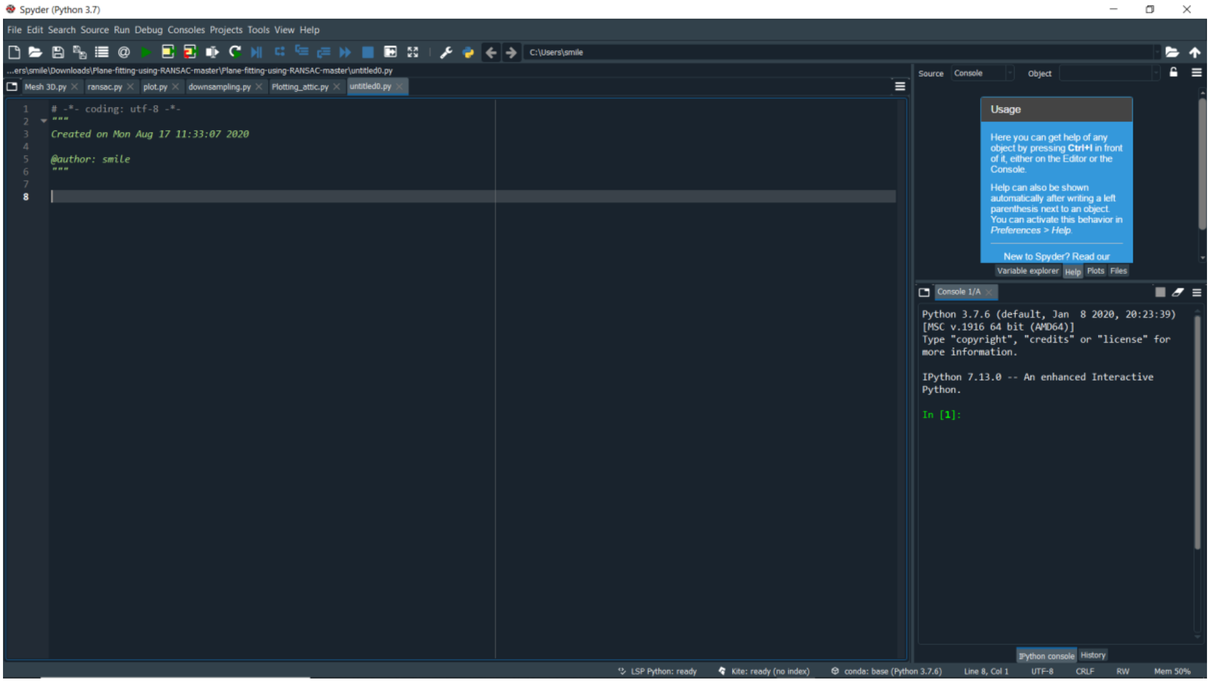
- Ransac algorithm operates based on the user defined parameters mainly distance threshold, number of inliers, trials and maximum probability [1][2][3][4][5][6].
- Iterations of Ransac algorithm needs to be considered for detecting the building planes [2].
- Combination of Ransac with statistical method and hypothesis provides higher probability for detecting good planes [1][3][4].
- Hardware components needs to be considered, which plays an important role for analysis and processing operations [5].
- Computer Vision algorithm with the use of correct parameters and data attributes provides an innovative path way for solving the real-world problems [1][2][3][4][5][6].

2.2 Semi-Automation Technique

The Semi-Automation technique is a programming-based approach which combines the use of powerful computer vision algorithms with computer machines for solving the real-world problems. The powerful and robust Ransac computer vision algorithm is used on point clouds to predict the planes and to segment the points from outliers and noises in-order to acquire a purified indoor point cloud environment. The methodology, workflow and limitations of the Semi-Automation technique is discussed detailly in Chapter 4.

2.2.1 Programming Interface

For powerful interaction of computer algorithms with the computer machine, an application programming interface needs to be present for bridging the gap between the man-made algorithms and computers. Spyder, an open source cross-platform integrated development environment which is used for scientific programming using python language toolkits and environment is used as a programming interface for this master thesis.



```
# -*- coding: utf-8 -*-
"""
Created on Mon Aug 17 11:33:07 2020

@author: smile
"""

```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39)
[MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for
more information.

IPython 7.13.0 -- An enhanced Interactive
Python.

In [1]:
```

Figure 3: Spyder Interface

2.2.2 Python Packages

For the master thesis research, the algorithms are completely written using Python language with the support and use of powerful python packages and libraries. The different types of python libraries are used for the Ransac algorithm based on the purpose and functionalities it performs and operates. The list of python libraries and their description are tabulated below:

Libraries	Description
NumPy	Used for n-dimensional arrays and numerical computing operations
Math	Used to perform mathematical functions and operations
Linear_Model	Perform linear regression operations
Matplotlib	Used to create static, animated and interactive visualization plots
Axes 3D	Used to create three-dimensional axis
Open3D	Perform 3D point cloud data processing, analysis and visualization
Ransac	Perform Ransac algorithm

Table 1: Python Libraries

2.2.3 Ransac Algorithm

Ransac stands for Random Sample Consensus is a robust iterative method used to estimate parameters of a mathematical model from a set of observed points which contains outliers and noises. The inlier points are detected and predicted using Least Square model (LSM) or Linear Regression Model (LRM) which is for 2D and for 3D operations, Planar Least Square Model or Planar Regression Model is used. The remaining points or outliers are extruded. The basic working principle of Ransac algorithm is described below:

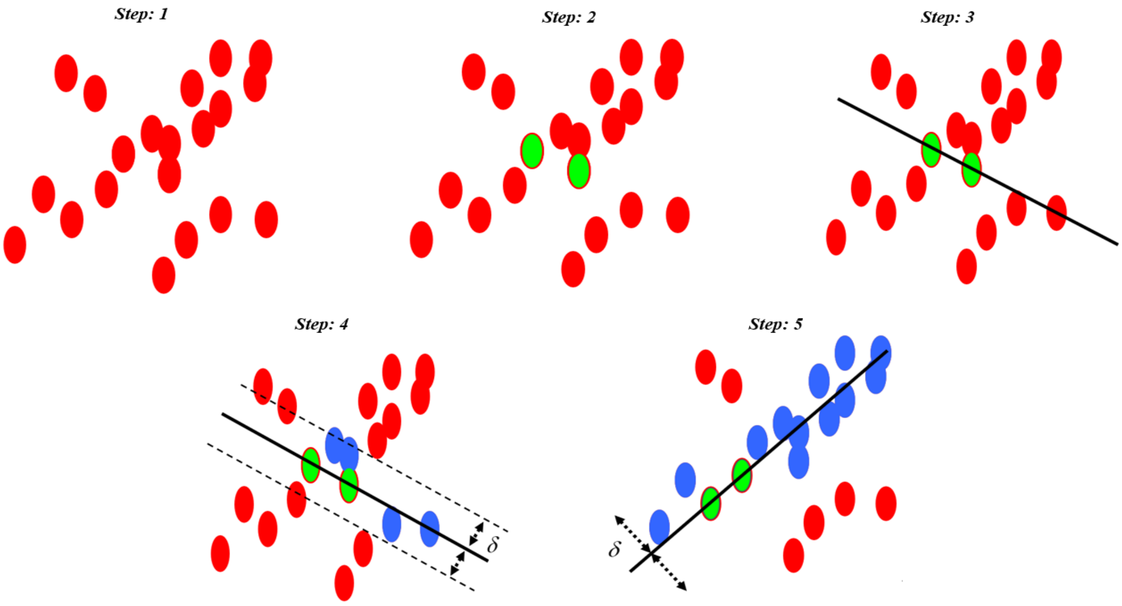


Figure 4: Ransac Algorithm (2D) [7]

- Step 1: Sample points are randomly distributed and required to fit a model.
- Step 2: Two random points are selected on first iteration.
- Step 3: Linear Regression model is fitted for the selected two random points.
- Step 4: Inlier Points are detected based on the user defined parameters such as distance threshold, trials, maximum probability etc.
- Step 5: Trials/Iterations are repeated again and again to attain the maximum probability of inliers.

The above steps describe the basic working principle of Ransac algorithm for 2-Dimensional points, whereas for the point clouds the 3-Dimensional parameters for the plane needs to be considered by defining the plane equation model to fit the 3D points.

$$Z = a*X + b*Y + d$$

For 3D, the plane equation model is fitted on the point clouds with the defined parameters for detecting the planes and to predict the inlier points that falls on the correct suitable planes. The correct suitable planes are predicted based on the iterations it operates and the parameters defined by the user.

2.3 Comparative Study

The major planes that are predicted using Ransac algorithm in semi-automation technique is compared with the external open-source automated software called CloudCompare (CC) which uses Ransac shape-detection plugin [16] for detecting major planes. The visual cross-validation is carried out on both techniques to find the number of major planes, correctness, outlier removal and to cross validate the result output intersection of two Ransac techniques. The comparative study is done in this research to check the correctness of the results of semi-automation technique and to provide a firm idea that the planes and points can be predicted using programming-based approach and not using the external software. The comparative plane studies are discussed detailly in Chapter 5 of the master thesis.

2.4 Automation Technique

The Automation technique is purely a software-based approach which uses in-built machine learning algorithms for data processing, analysing and visualizing the 3D point clouds. The automation technique involves a series of process for creating a purified mesh building model and an outlier mesh model. The powerful inbuilt machine learning algorithms present in the software helps to customize the input data and provides a valid meaningful output product based on this research task and purpose. The automation technique is discussed detailly in Chapter 6 of the master thesis.

2.4.1 Software Usage

For performing the automation technique, some of the valid relatable point cloud software's needs to be considered for attaining the research goal and objectives. The two major software's that contributed for automation technique are listed below:

2.4.1.1 LAS Tools

LAS Tools are the open source software suite which is a collection of highly efficient, batch scriptable, multicore command line tools for Lidar processing. LAS Tools consists of other tools to classify, tile, convert, filter, raster, split, merge, triangulate, contour, clip, polygonise the input Lidar data. The powerful software suite is used in this research for merging the continuous overlapping of point clouds. LAS tools provide an advantage such that the memory space requirement consumption is less and seamless processing of large point cloud data is easy to handle [8].

2.4.1.2 Agisoft Metashape

Agisoft Metashape is a stand-alone software product that performs photogrammetric operations for digital images and Lidar point cloud and generates 3D geo-spatial data models to be used in GIS application, cultural heritage documentation, visual effects production and indirect measurements of objects of various scales. The software uses powerful Machine learning and Computer Vision algorithms to produce high quality models and better results. It provides smart automated processing for high user interactivity [9]. The software is extensively used in this research for various tasks to create a high-quality data model. The tasks in this research includes the removal of outliers, filling of point cloud gaps and generation of high-quality 3D mesh model for purified building point cloud and outlier point clouds. The tasks are described briefly in Chapter 6 and sub-chapters of 6 of this master thesis.

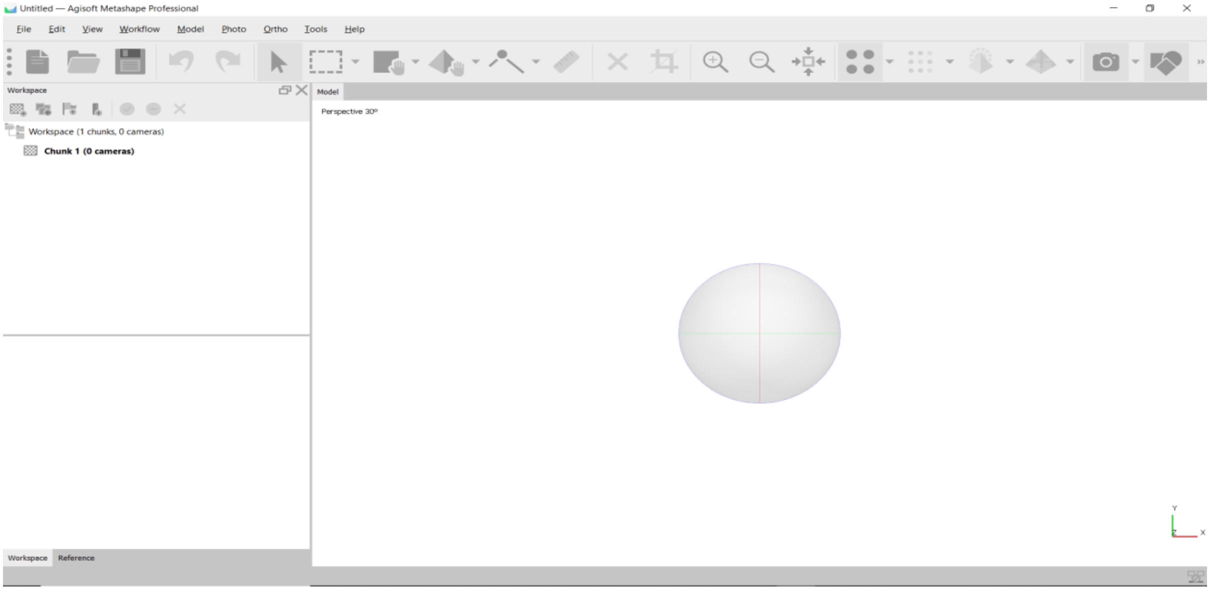


Figure 5: Agisoft Metashape Interface

2.5 Visualization Technique

For the master thesis, the visualization is one of the major important parts that provides a technique for interactively communicating the 3D mesh models and point clouds using web-based tools and technologies. This research provides a high interactivity of point clouds and 3D models with basic and advanced visualization functionalities and elements for the end user to play with the data products on web using opensource and commercial software products. The visualization of the output data products using web-based tools is discussed briefly in Chapter 7 of this master thesis.

2.5.1 Software Usage

For performing the high interactivity visualization with respect to the data products, the valid and relatable software's needs to be considered for performing the operations with high end functionalities. The software's that are selected to perform the high-end tasks are listed below:

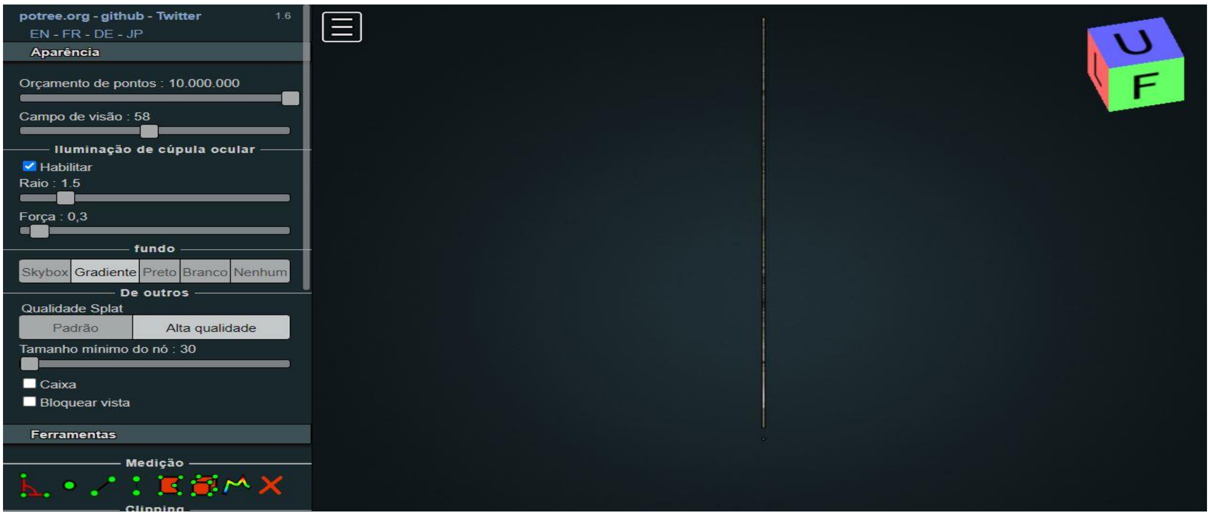
2.5.1.1 Sketchfab

Sketchfab is a commercial online platform to publish, share, discover, buy and sell 3D, AR and VR model contents. It provides a 3D model viewer with advanced functionalities and operations based on WebGL and WebVR technologies that allows the user to display 3D models on the web. The main advantage of Sketchfab is that the models can be visualized on any mobile browser, desktop browser and virtual reality headset [10]. For this research, the output 3D data mesh model contents are visualized individually on web with advanced visualization element functionalities and operations.

Figure 6: Sketchfab User Interface

2.5.1.2 Potree Converter

The Potree Converter is an open source online based point cloud viewer used for measuring, projecting and visualizing the point clouds on web. The Potree Converter normally converts point clouds to a format that is compatible with the Potree viewer. The Potree viewer helps to visualize the converted point clouds on web with high-end functionalities and operations [11]. For this research, Potree Converter is used to visualize the purified point clouds and outlier point clouds on web for the end user to interactively play with the data product with advanced visualization element functionalities and operations on web.


Orçamento de pontos : 10.000.000

Campo de visão : 58

☑ Habilitar

Raio : 1.5

Força : 0,3

Qualidade Splat

Tamanho mínimo do nó : 30

□ Caixa

□ Bloquear vista

Figure 7: Potree Web Viewer User Interface

3 DATA COLLECTION AND PREPARATION

The first sub-chapter describes the process involved for the collection of point cloud data which is followed by the second sub-chapter that describes how the collected point cloud data are prepared for the both semi-automation and automation techniques.

3.1 Collection of Point Cloud Data

For this research, the point clouds are taken as a base input product which is collected by using a terrestrial laser scanning instrument. The instrument scans the complete 3rd floor of the building with an overlap of 40-60% and a sidelap of 15-20%. The instrument uses high density scanning rate and records all the detailed information present in it. The point cloud data collected for this research study consists of 16 scans with high continuous overlapping precision and accuracy. The instrument also records the digital images of the scanned study area with a dynamic 360° view angle and static images. The point cloud data product contains the basic elements such as position of each point and their respective RGB [17].

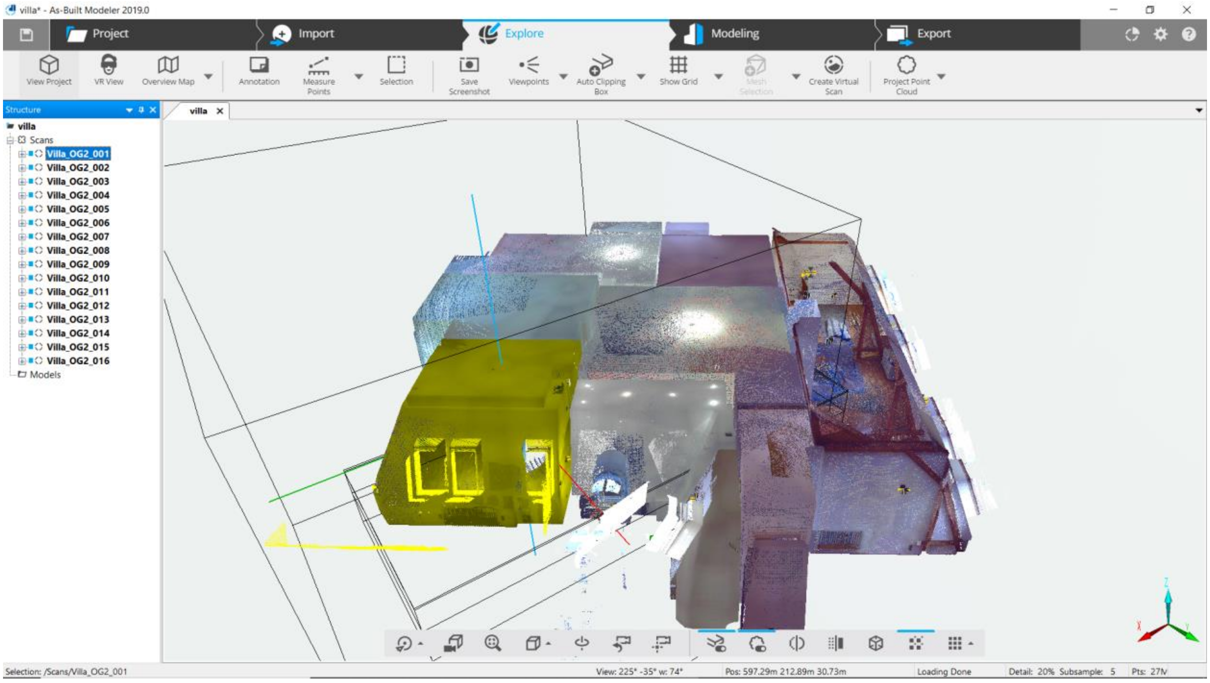


Figure 8: 16 Raw Scans of PCD

The collected 16 raw scans of point cloud data and images are visualized by using FARO As-Built Modeller software for quality checking, overlapping and precision. This base software is mainly used for preliminary visualization and investigation of the collected point cloud data.

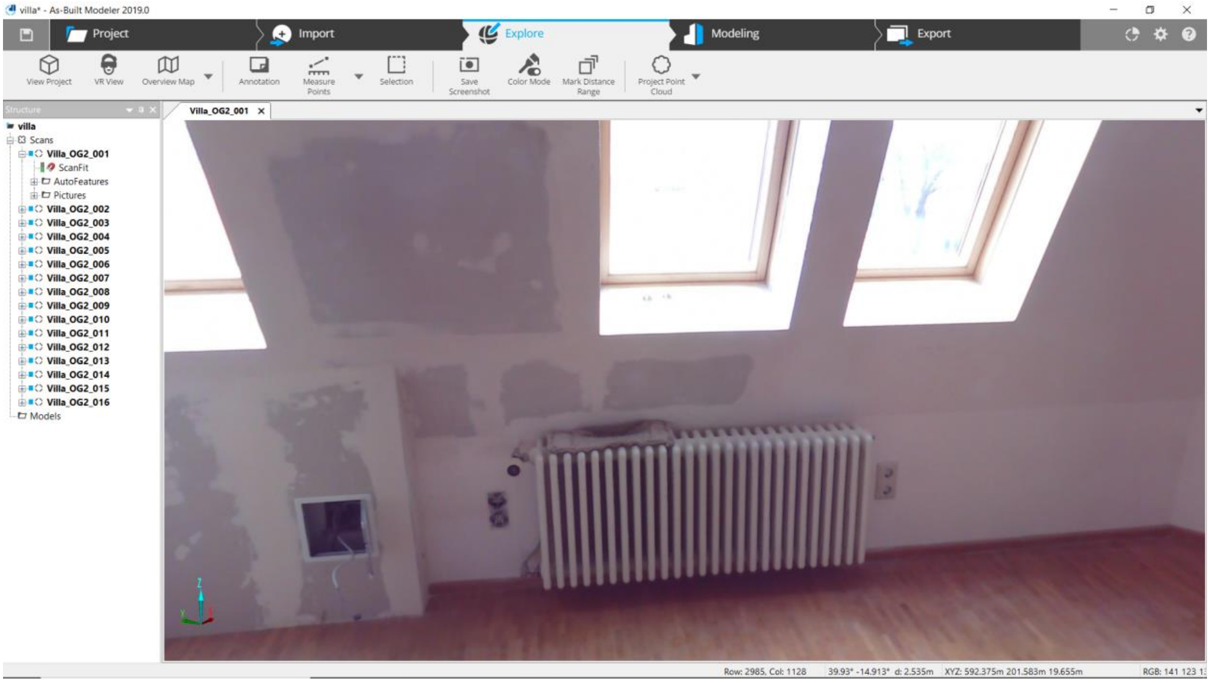


Figure 9: Sample Digital Image of Study area

3.2 Conversion of Raw Scans

The 16 raw scans are scanned, recorded and collected by FARO laser scanning instrument which has an extension file format called .fls (FARO Specific file format). For this research study, two important point cloud formats are considered and prepared by converting the .fls file format to a specific format that are used in both semi-automation and automation techniques. The important file formats considered are listed below:

3.2.1 PLY File Format

PLY format stands for Polygon File Format is a computer format which was designed to store 3-dimensional data from 3D scanners. This format supports a simple description of a single object as a list of nominally flat polygons [12] and stores 3D data as triangles. The PLY file format is used in this research for semi-automation technique, which is quite recognized and handled well by the python libraries [Table 1].

3.2.2 LAS File Format

LAS (LASer) format is a widely used file format, mainly designed for interchanging and archiving of 3-dimensional lidar point cloud data. LAS format is considered as an industry standard format for lidar data [13]. The LAS format is used in this research for the automation technique and is exceptionally handled well by the Point Cloud software's.

4 SEMI-AUTOMATION TECHNIQUE

This chapter discusses about one of the most important techniques called Semi-Automation technique, which is a programming-based approach that combines the use of powerful computer vision algorithms with computer machines for solving the real-world problems. The powerful robust Ransac computer vision algorithm is used on point clouds to predict the planes and to segment the points from outliers and noises (obstacles), in-order to acquire a purified indoor point cloud environment. The complete workflow techniques for two samples are discussed detailly in the following sub-chapters.

4.1 Methodology

The Semi-Automation workflow technique comprises a chain of tasks to acquire an output product of purified indoor point cloud environment. The tasks include the importing of point cloud samples into the programming interface [2.2.1] using correct and supporting python libraries [Table 1] for analysis and processing, implementation of Ransac algorithm on point clouds [2.2.3] followed by Ransac plane detection on sample points. The plane points are predicted and segmented from the samples, which automatically involves the process of outlier removal on point clouds. The above tasks are performed carefully to create a purified room geometry of the point cloud environment. The list of tasks after pre-processing is listed below as a graph figure:

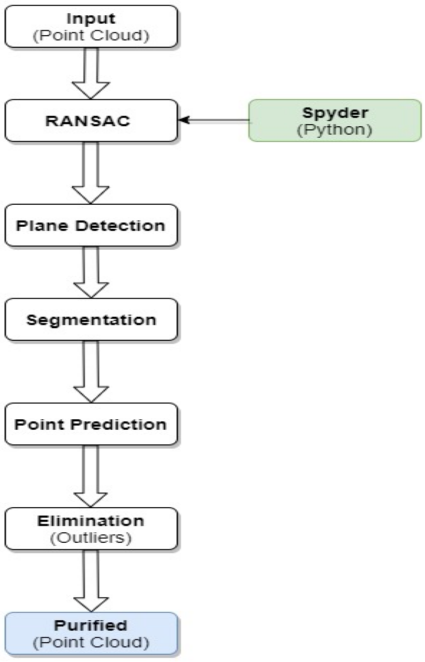


Figure 10: Semi-Automation Technique Methodology

4.2 Sample 1

For this research, One of the 16 scans are taken as a test subject and the semi-automation procedure is applied on it. Due to high cloud density of points and storage space, single scan is considered for the semi-automation technique. The sample test region is an attic type region which consists of windows, supporting beams, heaters, work floor mat etc present in it. The below sample is visualized using one of the point cloud python library called Open3D [Table 1] using the following python codes:

```
# Visualization of the point cloud data
import open3d as o3d
pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/Sample1.ply")
o3d.visualization.draw_geometries([pcd])
```

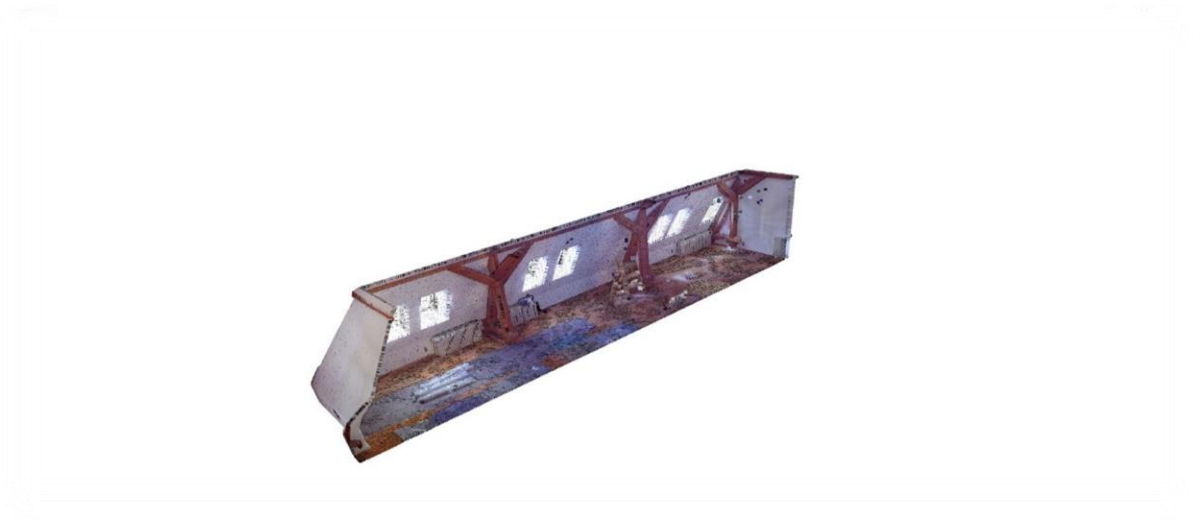


Figure 11: Sample 1

The technical specification details of sample 1 are tabulated below:

Technical Details	Description
No of Points	5,480,351
Density Rate	Very High
Texture	Includes Texture Colour (RGB)
Normals	Includes Vertex Normal (0,0,0)
File Format	PLY

Table 2: Sample 1 Specification

4.2.1 Down Sampled PCD

Due to high point cloud density and hardware specificity, the sample 1 is down-sampled to a voxel size of 0.2m. The down-sampled PCD is used as a test subject for the semi-automation technique. The meta data properties of the point cloud data remains same, whereas the quantity and quality of the point cloud data is altered. The shape and size geometry of the sample 1 remains same which still incudes the objects present in it. The below sample 1 is visualized and down-sampled using the following python libraries and codes:

```
# Down sampling of the point cloud data

import numpy as np
import open3d as o3d
pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/Sample1.ply")
o3d.visualization.draw_geometries([pcd])
downpcd = pcd.voxel_down_sample(voxel_size=0.2)
xyz = np.asarray(downpcd.points)
o3d.visualization.draw_geometries([downpcd])
```

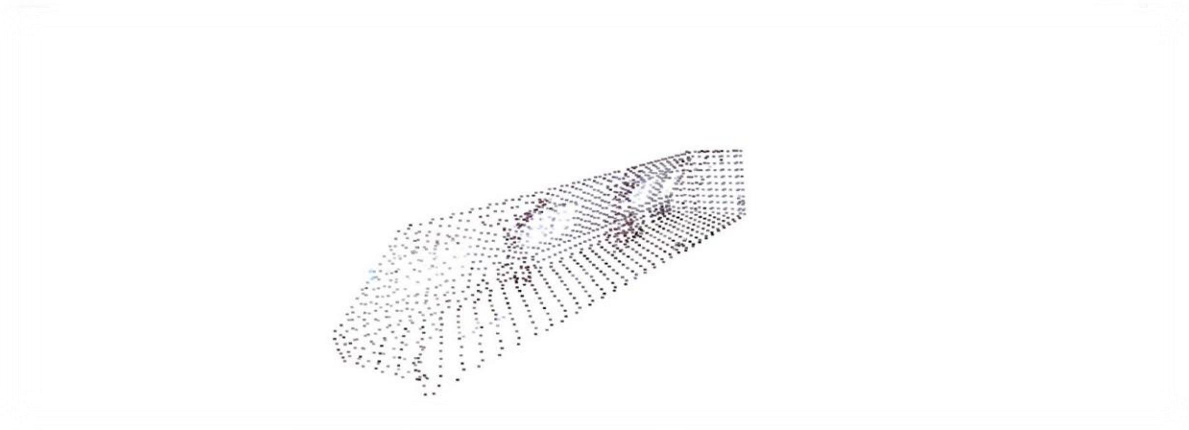


Figure 12: Down-sampled PCD (Sample 1)

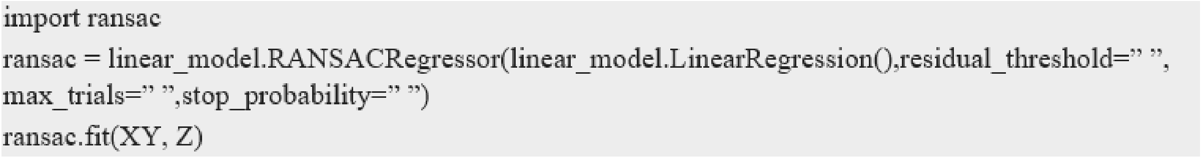
The technical specification details of down-sampled Sample 1 are tabulated below:

Technical Details	Description
No of Points	1,623
Density Rate	Sparse
Texture	Includes Texture Colour (RGB)
Normals	Includes Vertex Normal (0,0,0)
File Format	PLY

Table 3: Down-sampled Sample 1 Specification

4.2.2 Plane Fitting Using Ransac

In this research, the down-sampled PCD sample 1 is taken as a test subject and the robust Ransac algorithm is applied to it. The algorithm identifies and predicts the planes on the down-sampled PCD based on the user-defined parameters. The parameters such as linear regression model, residual threshold, iterations and stop to maximum probability of inliers are defined to the Ransac algorithm and the planes are predicted. The general syntax code for the Ransac algorithm are shown below:



```
import ransac
ransac = linear_model.RANSACRegressor(linear_model.LinearRegression(),residual_threshold=" ",
max_trials=" ",stop_probability=" ")
ransac.fit(XY, Z)
```

The parameters are the preliminary essentials that makes the algorithm a very complete one. For this research, the defined values for each parameter are listed below:

Parameters	Defined Value
Max_trials	20
Stop_probability	0.99
Residual_threshold	0.01

Table 4: Ransac Parameters (Sample 1)

The linear regression model is the most important parameter which defines a plane based on the assigned plane equation mentioned below:

$$Z = a*X + b*Y + d$$

The plane equation is defined by the angle of rotation, slope (a, b, d) and the data attributes (X, Y). For each iteration, the angle of rotation and slope values are predicted along with the data attributes for every Ransac plane. The linear regression model along with the above three mentioned parameters are defined in the Ransac algorithm to predict the suitable points that falls on the predicted suitable planes.

For identification of suitable planes, the algorithm is iterated again and again on the down-sampled Sample 1 to a maximum number of considered 20 iterations (user-defined) and the planes are identified and classified to the user-defined criteria. The below figure shows the 20 iterations on the down-sampled Sample 1:

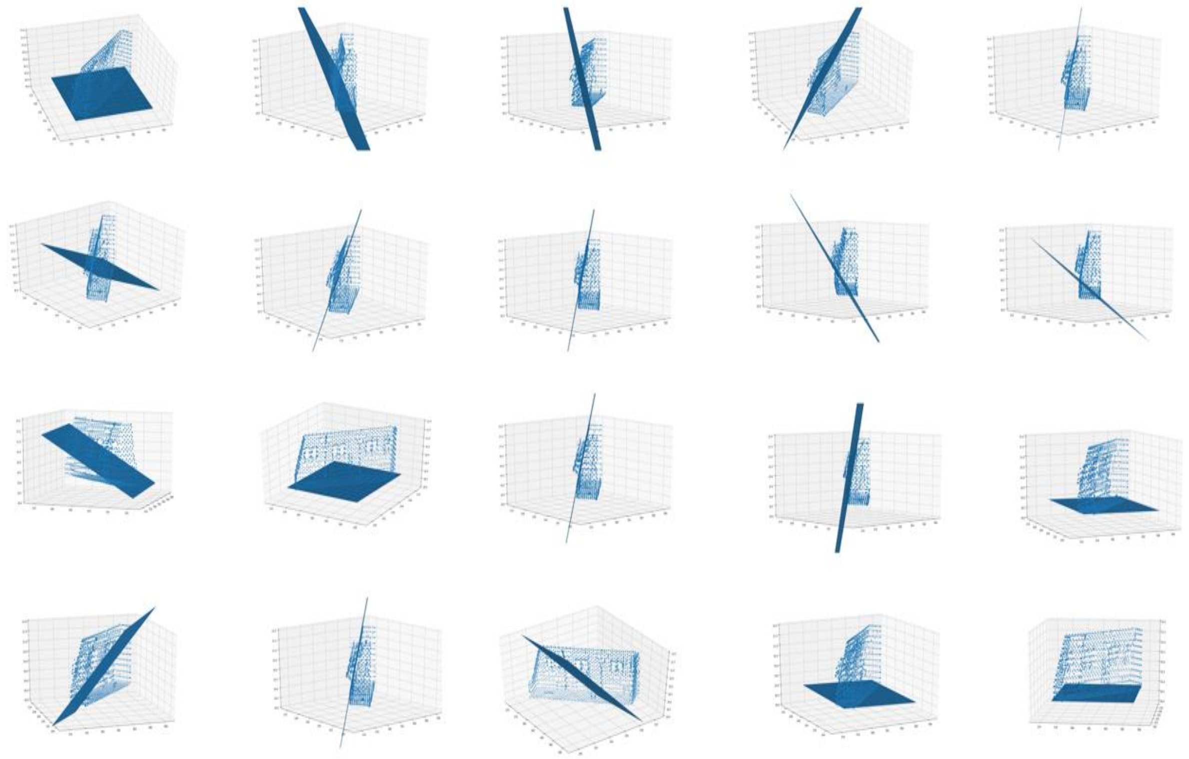


Figure 13: Iterations (Down-Sampled Sample 1)

4.2.3 Point Prediction and Segmentation

For each iteration, the Ransac plane that falls on the down-sampled sample 1, the points are predicted and segmented. The suitable planes are considered and the points are predicted based on the defined criteria. The iterations considered for identifying the suitable planes and for predicting and segmenting the points are tabulated below:

Research Defined Criteria	Iterations
High Favourable Planes	1, 5, 8, 13, 17
Medium Favourable Planes	7, 12, 15, 19, 20
Low Favourable Planes	2, 3, 4, 6, 9, 10, 11, 14, 16, 18

Table 5: Plane Criteria (Down-Sampled Sample 1)

For this research, only high favourable planes are considered for point prediction. The points are predicted based on the Z values that are identified for each and every point which falls on high favourable planes. The identified Z values are cross correlated with the main down-sampled data Z values for identifying the corresponding X and Y values, in-order to segment the suitable points. The Z values also provide the range of co-ordinates data in which the set of co-ordinates that are need to be focused and considered for predicting the points. The below graph shows the predicted Z values for 20 iterations on down-sampled sample 1:

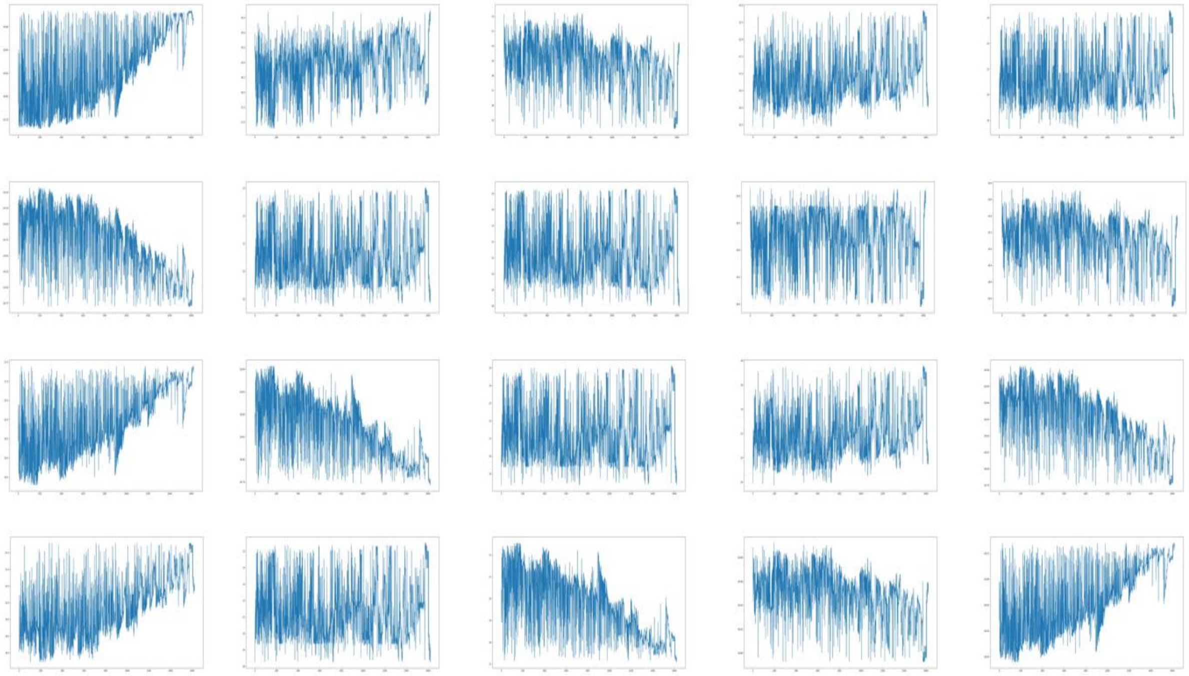
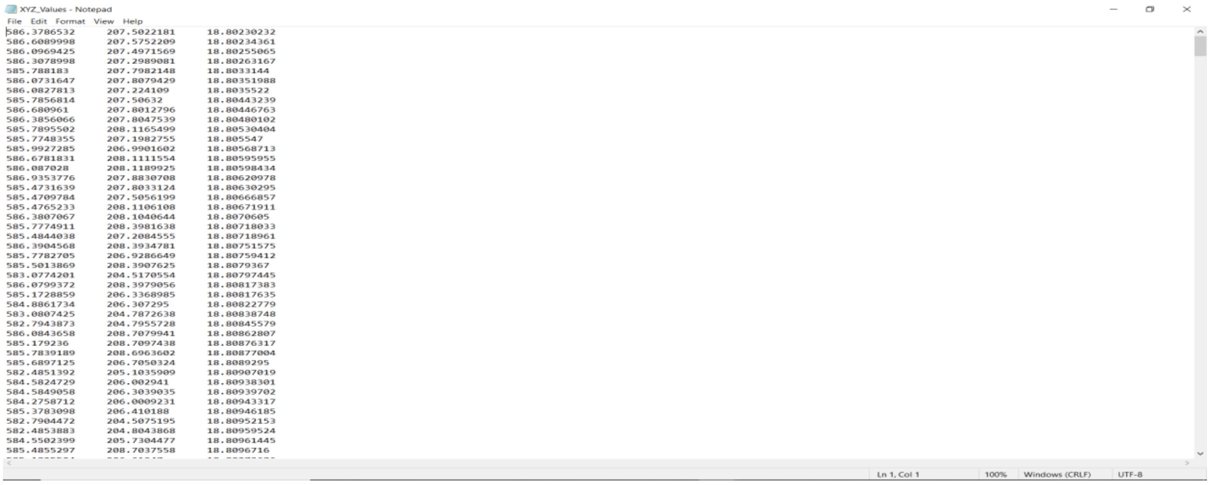


Figure 14: Z values (Down-Sampled Sample 1)

After Cross-Correlation, the predicted points (X, Y, Z) are stored and saved as a notepad document in the form of text file format for visualization. The predicted points are shown below in the form of text figure:



```
586.3786532      207.5022181      18.80230232
586.6089998      207.5752209      18.80234361
586.0969425      207.4971569      18.80255065
586.3078998      207.2989081      18.80263167
585.788183       207.7982148      18.8033144
586.0731647      207.8079429      18.80351988
586.0827813      207.224109       18.8035522
585.7856814      207.50632        18.80443239
586.680961       207.8012796      18.80446763
586.3856066      207.8047539      18.80480102
585.7895502      208.1165499      18.80530404
585.7748355      207.1982755      18.805547
585.9927285      206.9901602      18.80568713
586.6781831      208.1111554      18.80595955
586.087028       208.1189925      18.80598434
586.9353776      207.8830708      18.80620978
585.4731639      207.8033124      18.80630295
585.4709784      207.5056199      18.80666857
585.4765233      208.1106108      18.80671911
586.3807067      208.1040644      18.8070605
585.7774911      208.3981638      18.80718033
585.4844038      207.2084555      18.80718961
586.3904568      208.3934781      18.80751575
585.7782705      206.9286649      18.80759412
585.5013869      208.3907625      18.8079367
583.0774201      204.5170554      18.80797445
586.0799372      208.3979056      18.80817383
585.1728859      206.3368985      18.80817635
584.8861734      206.307295       18.80822779
583.0807425      204.7872638      18.80838748
582.7943873      204.7955728      18.80845579
586.0843658      208.7079941      18.80862807
585.179236       208.7097438      18.80876317
585.7839189      208.6963602      18.80877004
585.6897125      206.7050324      18.8089295
582.4851392      205.1035909      18.80907019
584.5824729      206.002941       18.80938301
584.5849058      206.3039035      18.80939702
584.2758712      206.0009231      18.80943317
585.3783098      206.410188       18.80946185
582.7904472      204.5075195      18.80952153
582.4853883      204.8043868      18.80959524
584.5502399      205.7304477      18.80961445
585.4855297      208.7037558      18.8096716
```

Figure 15: Point Prediction (Down-Sampled Sample 1)

4.2.4 Removal of Outliers

In this research, the selection of high favourable planes leads to the maximum dominant removal of outliers. The high favourable planes consist of two major planes which includes the inclined wall of the attic area and the floor region. The outliers such as working ladder, windows, supporting beams, floor mats and heaters are removed. Due to correct and valid selection of distance/residual threshold value parameter [Table 4] the inliers (wall and floor) are predicted with high suppression of obstacle noises. The below figure represents the outliers that are present in the down-sampled sample 1 with an enclosed red bounding box:

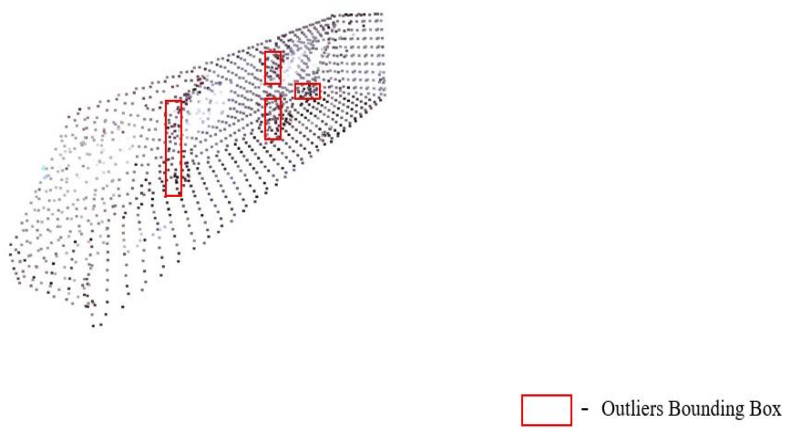


Figure 16: Outliers (Down-Sampled Sample 1)

4.2.5 Purified PCD

After the implementation of Ransac algorithm and prediction of suitable planes and points based on the defined parameters, the outliers or obstacles are removed for the attic region and a purified room geometry of indoor point cloud is created. The purified room geometry consists of only two major planes which includes the inclined wall of the attic region and the floor area. The predicted points from the down-sampled point clouds are reduced in number due to the exclusion of the noises and obstacles. The predicted points are reduced to a number count of 910 points which includes two major geometry regions (wall and floor). The below figure represents the purified indoor room geometry of point clouds:

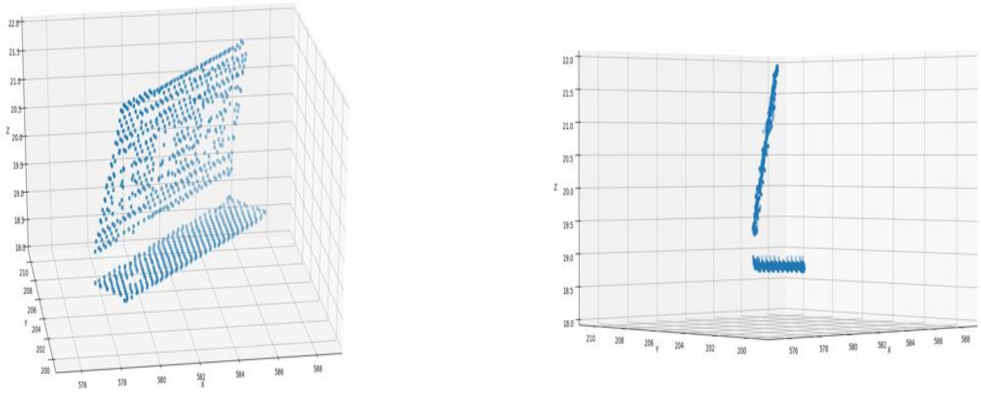


Figure 17: Purified Sample 1

4.3 Sample 2

For this research, one of the 15 scans is taken as a second sample test subject and the semi-automation technique is applied to it. Due to storage space limit and very high point cloud density, single scan is considered for the semi-automation technique. The second sample test region is a single box shaped room which consists of windows, heaters and other work objects on the floor such as paint bucket and working floor mat etc. The second sample is mainly considered for this research to interpret and evaluate the Ransac algorithm for a closed region which consists of vertical walls and horizontal floors. The second sample is visualized below using one of the python point cloud library called Open3D [Table 1] using the following python codes:

```
# Visualization of the point cloud data
import open3d as o3d
pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/Sample2.ply")
o3d.visualization.draw_geometries([pcd])
```

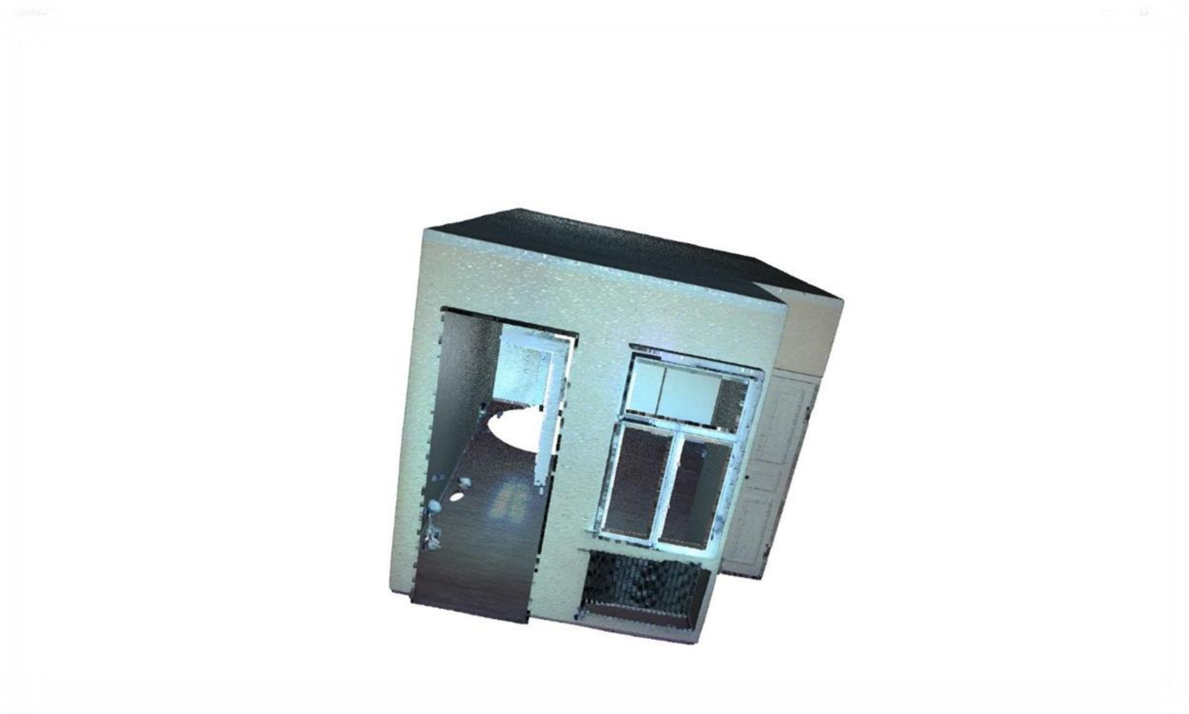


Figure 18: Sample 2

The technical meta-data specification details of Sample 2 are tabulated below:

Technical Details	Description
No of Points	40,549,076
Density Rate	Very High
Texture	Includes Texture Colour (RGB)
Normals	Includes Vertex Normal (0,0,0)
File Format	PLY

Table 6: Sample 2 Specification

4.3.1 Down Sampled PCD

Due to very high density of points and hardware specificity, the sample 2 is down-sampled to a voxel size of 0.2m. The down-sampled sample 2 is used as a test subject for this semi-automation research. The meta data properties and the geometric properties (shape & size) of the test subject remains same whereas the quantity and quality of the test subject is altered. The obstacles/noises remain same in the room. The below sample 2 is down-sampled and visualized using the following libraries and codes:

```
# Down sampling of the point cloud data
import numpy as np
import open3d as o3d
pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/Sample2.ply")
o3d.visualization.draw_geometries([pcd])
downpcd = pcd.voxel_down_sample(voxel_size=0.2)
xyz = np.asarray(downpcd.points)
o3d.visualization.draw_geometries([downpcd])
```

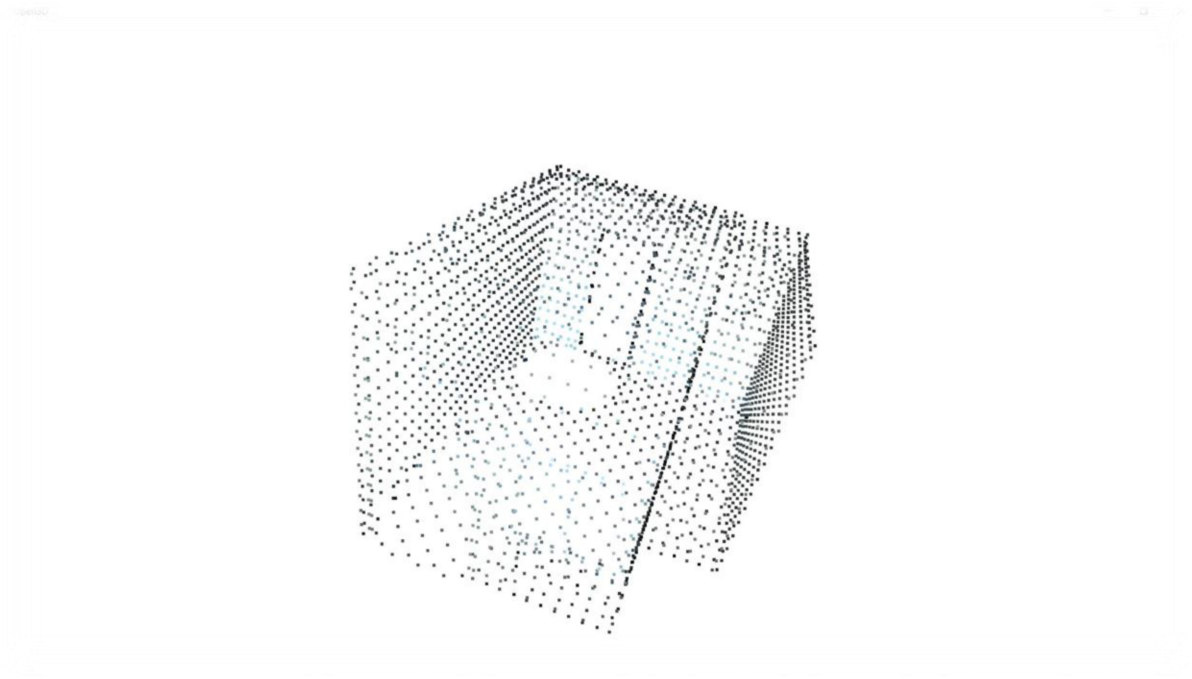


Figure 19: Down-Sampled PCD (Sample 2)

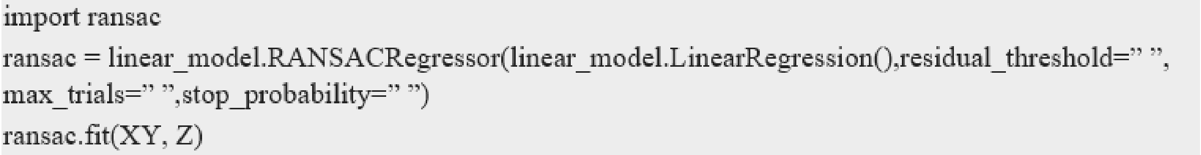
The technical meta-data specification details of down-sampled Sample 2 are tabulated below:

Technical Details	Description
No of Points	2,524
Density Rate	Sparse
Texture	Includes Texture Colour (RGB)
Normals	Includes Vertex Normal (0,0,0)
File Format	PLY

Table 7: Down-Sampled Sample 2 Specification

4.3.2 Plane Fitting Using Ransac

In this research, the down-sampled PCD sample 2 is taken as a second test subject and the powerful robust Ransac computer vision algorithm is applied to it. The Ransac algorithm identifies and predicts the planes on the down-sampled sample 2 based on the defined parameters. The parameters such as linear regression model, residual threshold, maximum trials and inliers probability are defined to the algorithm and the planes are predicted. The general syntax code for the Ransac algorithm are shown below:



```
import ransac
ransac = linear_model.RANSACRegressor(linear_model.LinearRegression(),residual_threshold=" ",
max_trials=" ",stop_probability=" ")
ransac.fit(XY, Z)
```

For this research, the defined values for each parameter of Sample 2 are tabulated below:

Parameters	Defined Value
Max_trials	20
Stop_probability	0.99
Residual_threshold	0.01

Table 8: Ransac Parameters (Sample 2)

The linear regression model is the most important parameter which defines a plane based on the assigned plane equation mentioned below:

$$Z = a*X + b*Y + d$$

The plane equation is defined by the angle of rotation, slope (a, b, d) and the data attributes (X, Y). For each iteration, the angle of rotation and slope values are predicted along with the data attributes for every Ransac plane. The linear regression model along with the above three mentioned parameters are defined in the Ransac algorithm to predict the suitable points that falls on the predicted suitable planes.

For identification of suitable planes, the algorithm is iterated again and again on the down-sampled Sample 2 to a maximum number of considered 20 iterations (user-defined) and the planes are identified and classified to the user-defined criteria. The below figure shows the 20 iterations on the down-sampled Sample 2:

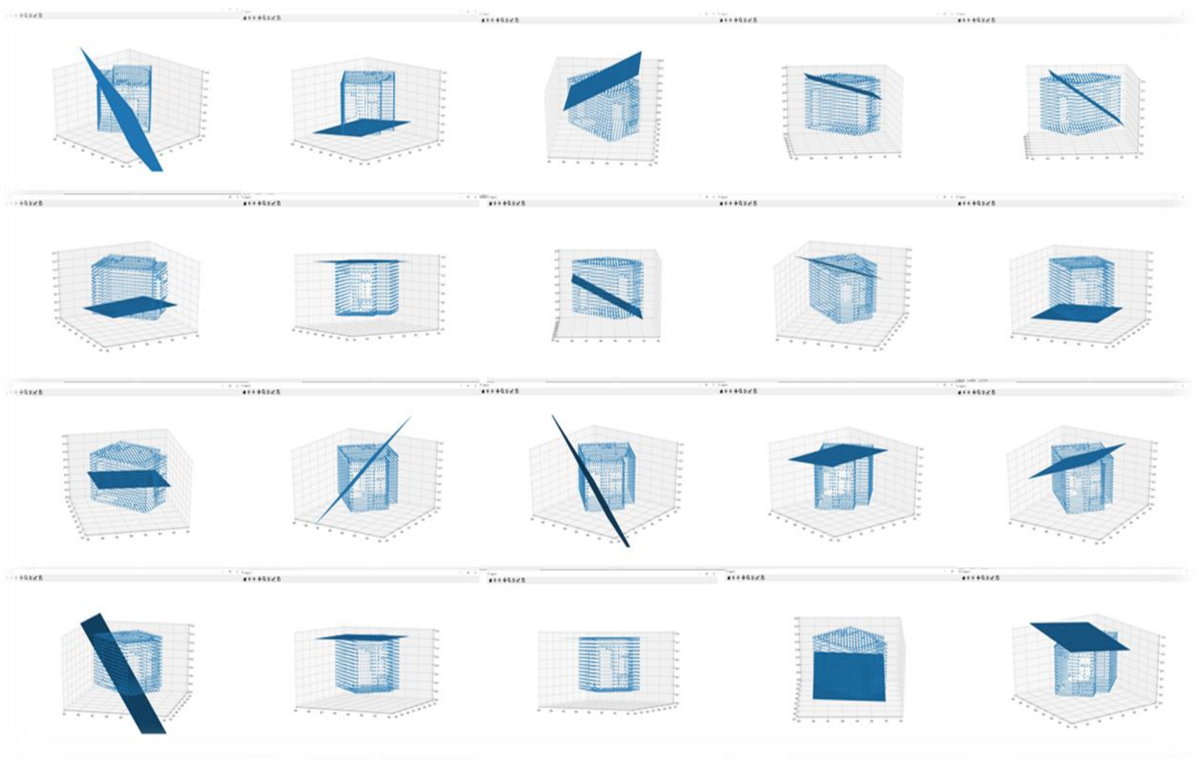


Figure 20: Iterations (Down-Sampled Sample 2)

RQ.1 - What kind of algorithm and model will be used for plane fitting on 3D points?

The powerful robust Ransac algorithm and linear regression model are used for plane fitting on 3D points for both down-sampled sample 1 and sample 2 point cloud data in the semi-automation technique.

4.3.3 Point Prediction and Segmentation

For each iteration, the Ransac plane that falls on the down-sampled sample 2, the points are predicted and segmented. The suitable planes are considered and the points are predicted based on the defined criteria. The iterations considered for identifying the suitable planes and for predicting and segmenting the points are tabulated below:

Research Defined Criteria	Iterations
High Favourable Planes	7, 10, 17, 18, 20
Medium Favourable Planes	2, 6, 14
Low Favourable Planes	1, 3, 4, 5, 8, 9, 11, 12, 13, 15, 16, 19

Table 9: Plane Criteria (Down-Sampled Sample 2)

For this research, only high favourable planes are considered for point prediction. The points are predicted based on the Z values that are identified for each and every point which falls on high favourable planes. The identified Z values are cross correlated with the main down-sampled data Z values for identifying the corresponding X and Y values, in-order to segment the suitable points. The Z values also provide the range of co-ordinates data in which the set of co-ordinates that are need to be focused and considered for predicting the points. The below graph shows the predicted Z values for 20 iterations on down-sampled sample 2:

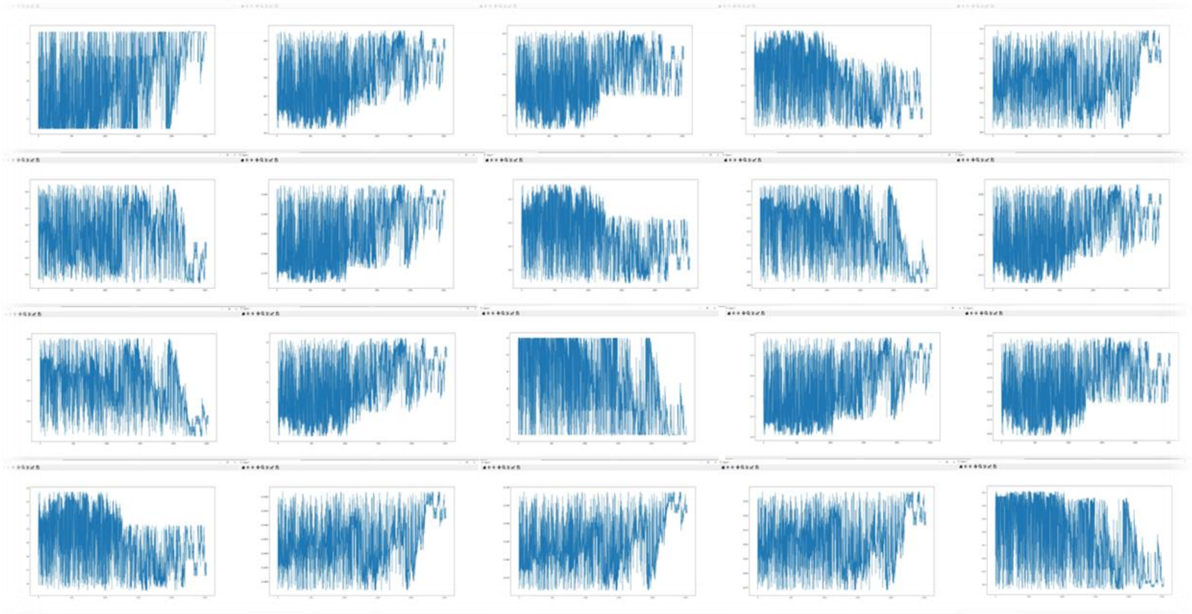
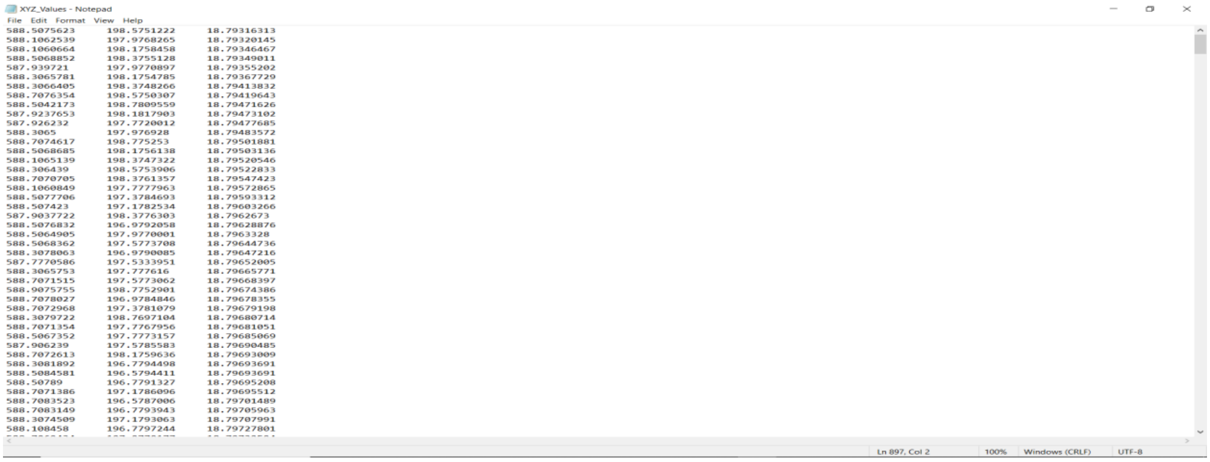


Figure 21: Z values (Down-Sampled Sample 2)

After Cross-Correlation, the points (X, Y, Z) are predicted from the down-sampled data and is saved as a notepad text file document for visualization. The predicted points are shown below in the form of text figure:



```
588.5075623     198.5751222     18.79316313
588.1062539     197.9768265     18.79320145
588.1060664     198.1758458     18.79346467
588.5068852     198.3755128     18.79349011
587.939721      197.9770897     18.79355202
588.3065781     198.1754785     18.79367729
588.3066405     198.3748266     18.79413832
588.7076354     198.5750307     18.79419643
588.5042173     198.7809559     18.79471626
587.9237653     198.1817903     18.79473102
587.926232      197.7720012     18.79477685
588.3065        197.976928      18.79483572
588.7074617     198.775253      18.79501881
588.5068685     198.1756138     18.79503136
588.1065139     198.3747322     18.79520546
588.306439      198.5753906     18.79522833
588.7070705     198.3761357     18.79547423
588.1060849     197.7777963     18.79572865
588.5077706     197.3784693     18.79593312
588.507423      197.1782534     18.79603266
587.9037722     198.3776303     18.7962673
588.5076832     196.9792058     18.79628876
588.5064905     197.9770001     18.7963328
588.5068362     197.5773708     18.79644736
588.3078063     196.9790085     18.79647216
587.7770586     197.5333951     18.79652005
588.3065753     197.777616      18.79665771
588.7071515     197.5773062     18.79668397
588.9075755     198.7752901     18.79674386
588.7078027     196.9784846     18.79678355
588.7072968     197.3781079     18.79679198
588.3079722     198.7697104     18.79680714
588.7071354     197.7767956     18.79681051
588.5067352     197.7773157     18.79685069
587.906239      197.5785583     18.79690485
588.7072613     198.1759636     18.79693009
588.3081892     196.7794498     18.79693691
588.5084581     196.5794411     18.79693691
588.50789       196.7791327     18.79695208
588.7071386     197.1786096     18.79695512
588.7083523     196.5787006     18.79701489
588.7083149     196.7793943     18.79705963
588.3074509     197.1793063     18.79707991
588.108458      196.7797244     18.79727801
```

Ln 897, Col 2   100%   Windows (CRLF)   UTF-8

Figure 22: Point Prediction (Down-Sampled Sample 2)

RQ.2 – What are the different criteria to make a plane wall plane?

Based on the suitable selection of iterations, the different criteria are considered for this research to make a wall plane are

- High Favourable Planes
- Medium Favourable Planes
- Low Favourable Planes

RQ.3 – What kind of plane points will be extruded in the semi-automated technique?

In this research, only high favourable planes with suitable iterations are considered for both down-sampled sample 1 and sample 2. Due to high removal of noises/obstacles, the medium and low favourable plane points are neglected/extruded in the semi-automation technique.

4.3.4 Removal of Outliers

In this research, the selection of high favourable planes leads to the maximum removal of outliers. The high favourable planes consist of two major planes which includes the ceiling area and the floor region. Due to the limitation [4.4] of the defined Ransac algorithm, the vertical 90° wall planes are difficult to predict and segment the suitable points from outliers. The outliers on the floor region such as working mat, paint bucket and other obstacles are removed. Due to correct and valid selection of distance/residual threshold value parameter [Table 4] the inliers (floor and ceiling) are predicted with high compression of noises/obstacles with the exclusion of vertical wall planes. The below figure represents the outliers that are present in the down-sampled sample 2 with an enclosed red bounding box:

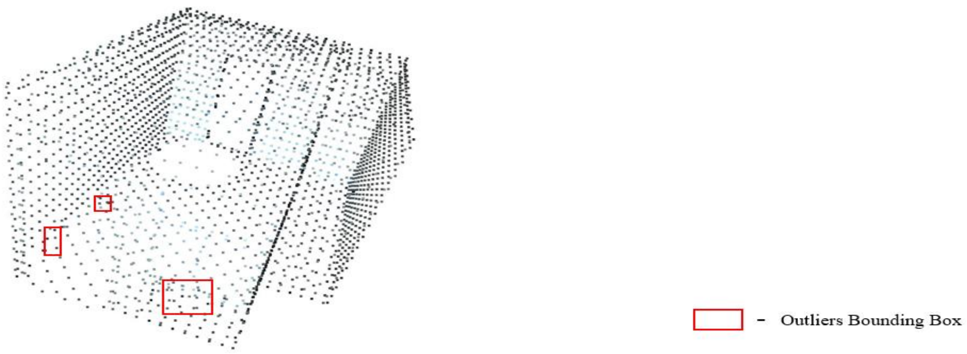


Figure 23: Outliers (Down-Sampled Sample 2)

4.3.5 Purified PCD

After the implementation of Ransac algorithm and prediction of suitable planes and points based on the defined parameters, the outliers are removed for the ceiling area and the floor region and a purified room geometry of indoor point cloud is created with an exclusion of predicting the vertical wall planes. Due to the limitation [4.4] of the defined Ransac algorithm, the vertical 90° wall planes are difficult to predict and segment the suitable points from outliers such as heaters and windows attached to the walls. For this Sample 2, only two major planes are predicted which includes the ceiling area and the floor region. The predicted horizontal points from the down-sampled point clouds are reduced in number due to the exclusion of noises and obstacles. The predicted points are reduced to a number count of 897 points which includes two geometry regions (Ceiling and floor). The below figure represents the purified indoor room geometry of point clouds:

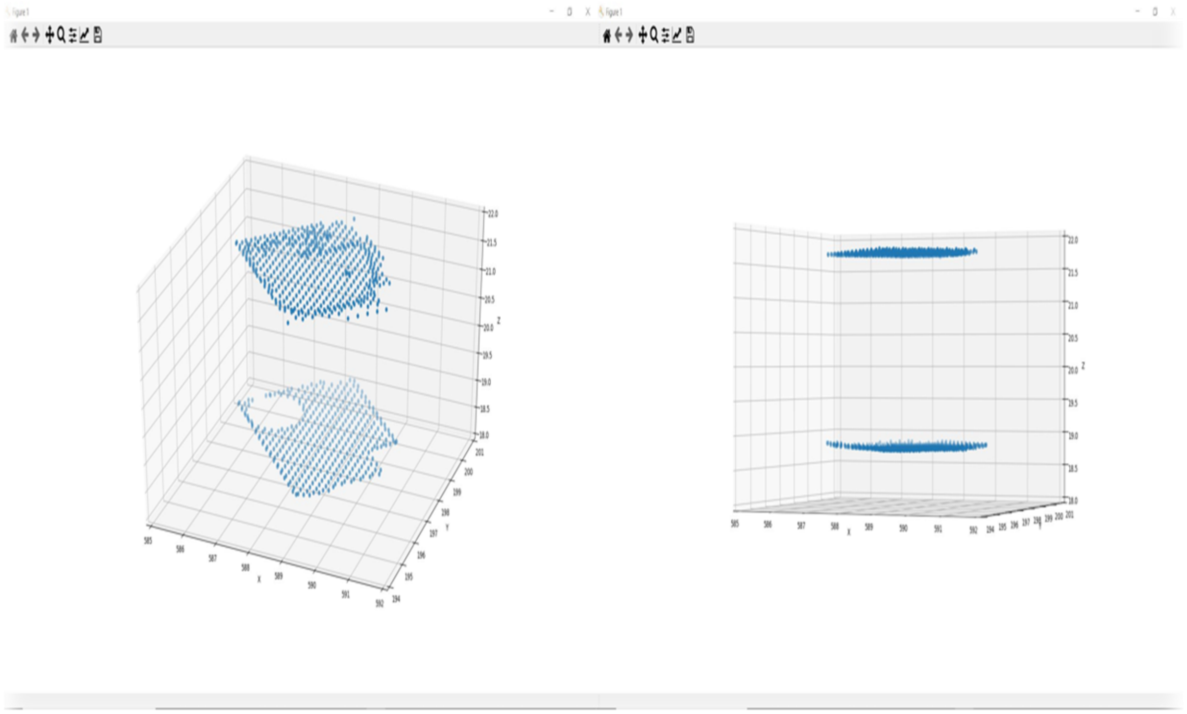


Figure 24: Purified Sample 2

4.4 Limitations of Semi-Automation Technique

In this research, the use of defined Ransac algorithm has some limitations on Semi-Automation technique. The Ransac algorithm defined in this research, works explicitly for inclined and horizontal planes of Sample 1, whereas for the vertical wall planes of Sample 2 are difficult to predict and segment the points. The limitations of the defined Ransac algorithm in this master thesis are listed below

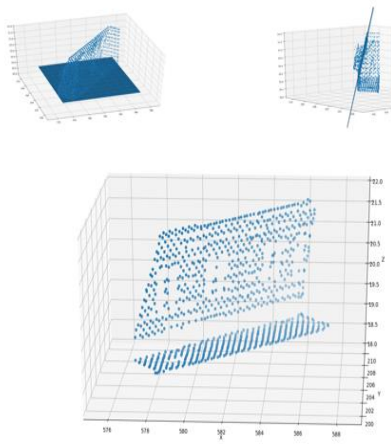
- The Ransac planes are difficult to predict for the 90° vertical walls. The algorithm uses arctan2 math for the plane angle, but failed to predict vertical geometry planes (walls).
- Each iteration for Ransac algorithm is done manually, due to the absence of loop concept in the algorithm.
- Each iteration can fit only one single Ransac plane on the Point cloud data.
- Multiple plane fitting on the point cloud data using the defined Ransac algorithm is not possible.

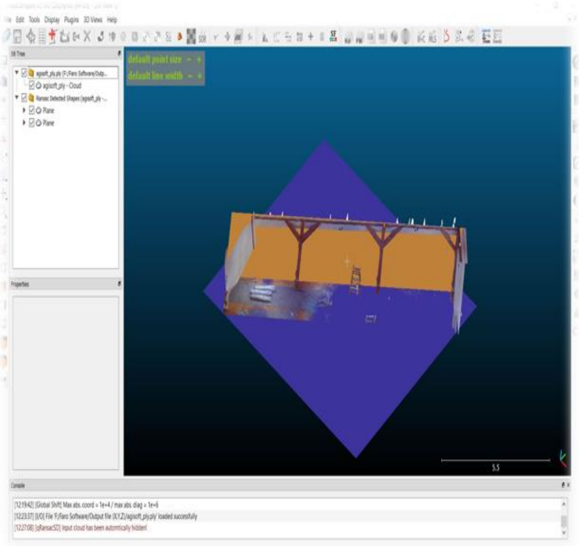
5 COMPARATIVE STUDY

In this chapter, the predicted Ransac planes and points using semi-automation technique [Chapter 4] are compared with an external automated software called CloudCompare (CC), which uses Ransac shape-detection plugin for predicting the major planes. In this research, the comparative study analysis is done to evaluate and to interpret the correctness of the semi-automation technique over automated software. For evaluation and interpretation, semi-automation technique-based sample 1 and sample 2 results are visually cross-validated with the automated software-based output products of same samples. The sub-chapters of both sample 1 and sample 2 describes detailly about the comparative study analysis that has been carried out in this master thesis.

5.1 Sample 1 Comparative Plane Analysis

For sample 1, the planes are predicted using a defined Ransac algorithm and the points that fall on suitable planes are predicted and segmented based on the defined parameters. The semi-automation technique is a programming-based approach, such that the results will purely depend on the defined algorithm and supporting parameters. The CloudCompare is an opensource point cloud viewer and analyser, which uses powerful machine learning and computer vision algorithms for analysing and solving the real-world problems. The sample 1 is inputted into both of these techniques and the results that follow in this research are absolute, promising and accurate to a maximum extent. The below figure shows the output of major planes that are predicted using python and CloudCompare:





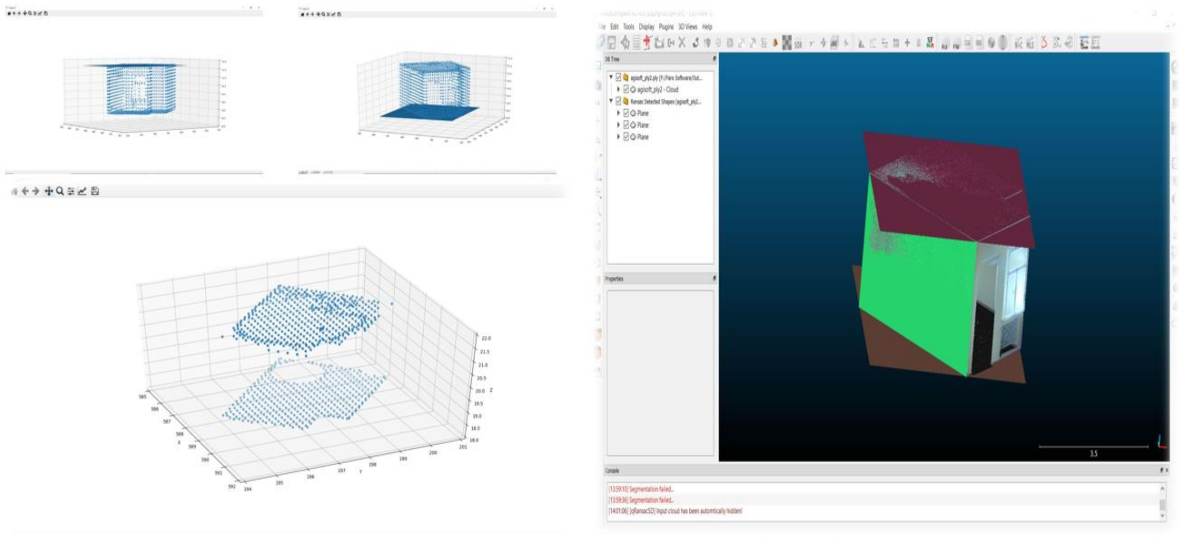
```
[12:19:42] [Global Shift] Max abs. coord = 1e+4 / max abs. diag = 1e+6
[12:23:37] [I/O] File 'F:/Faro Software/Output file (X,Y,Z)/agisoft_ply.ply' loaded successfully
[12:27:08] [qRansacSD] Input cloud has been automatically hidden!
```

Figure 25: Sample 1 Comparative Plane Analysis (Left: Python & Right: CC)

On visual cross-validation of both sample 1 results, two dominant planes are found. The plane regions include the inclined wall of an attic area and the floor region. The down-sampled sample 1 in python consists of predicted points from suitable and high major planes, whereas in CloudCompare the major planes are predicted using Ransac shape-detection plugin (default parameters for planes). The planes that fall on the point clouds in CloudCompare remains same with the semi-automation technique. The outliers that are present in CloudCompare are supporting beams, ladder, windows and working floor mat etc are removed completely to a dominant level in python and a purified room geometry is created with the maximum removal of outliers. The defined Ransac algorithm and parameters contribute to the high removal of outliers in python. The visual cross validation of the output results gives 100% intersection of correctness and accuracy of defined python algorithm on point clouds over automated technique.

5.2 Sample 2 Comparative Plane Analysis

For Sample 2 research, the planes and points that are predicted and segmented based on the defined Ransac algorithm and user-defined parameters in semi-automation technique is visually cross validated with the automated CloudCompare (CC) software. The sample 2 is fed into both of these techniques and the results that follow in this research are not quite promising and absolute, due to the limitations of the defined Ransac algorithm [Sub-chapter 4.4]. The below figure shows the output of major planes that are predicted using Python and CloudCompare (CC):



```
[13:59:10] Segmentation failed..
[13:59:36] Segmentation failed..
[14:01:06] [qRansacSD] Input cloud has been automtically hidden!
```

Figure 26: Sample 2 Comparative Plane Analysis (Left: Python & Right: CC)

On visual cross-validation of both sample 2 results, two major planes are found using semi-automation technique. Due to the limitations of the defined Ransac algorithm [Sub-chapter 4.4], the vertical dominant plane walls are not predicted and segmented. The plane regions that are found using python are the ground floor area and the ceiling region which leads to the 50% removal of outliers on the horizontal area section, whereas for the obstacles such as windows that are present on the vertical area section are not eliminated. The CloudCompare predicts the two horizontal major planes as well as the two vertical side wall planes which leads to the high dominant removal of outliers. The visual cross-validation of the output results gives 50% intersection of correctness and accuracy of defined python algorithm on point clouds over automated technique.

6 AUTOMATION TECHNIQUE

This chapter discusses about another important technique called automation technique, which is a software-based approach that uses in-built machine-learning and computer vision algorithms for data processing, analysing and visualizing the 3D point clouds. The use of powerful inbuilt algorithms helps to customize the data input and provides a valid output product based on the research task and objectives. The automation technique involves a serious list of multiple tasks to create a purified indoor point cloud environment. The following sub-chapters describes detailly about the creation of purified indoor point cloud geometry and purified point cloud model.

6.1 Methodology

The Automation workflow technique comprises of chain of tasks to acquire an output products of purified indoor point cloud geometry and purified point cloud model. The use of valid and suitable software's in this technique leads to the high proximity of correctness and precision of the output data products. The tasks include the importing of point cloud data into the software interface, which is followed by merging of overlapping samples using LAS Merge. The merged point cloud data is processed, analysed and visualized as a purified point cloud data and model using Agisoft Metashape software. The main objective of this research is to create purified mesh model for both buildings and outliers that are present in it, from a purified indoor point cloud geometry. The above research tasks are performed carefully to meet the research objective goal of the master thesis. The list of tasks is listed below as a graph figure:

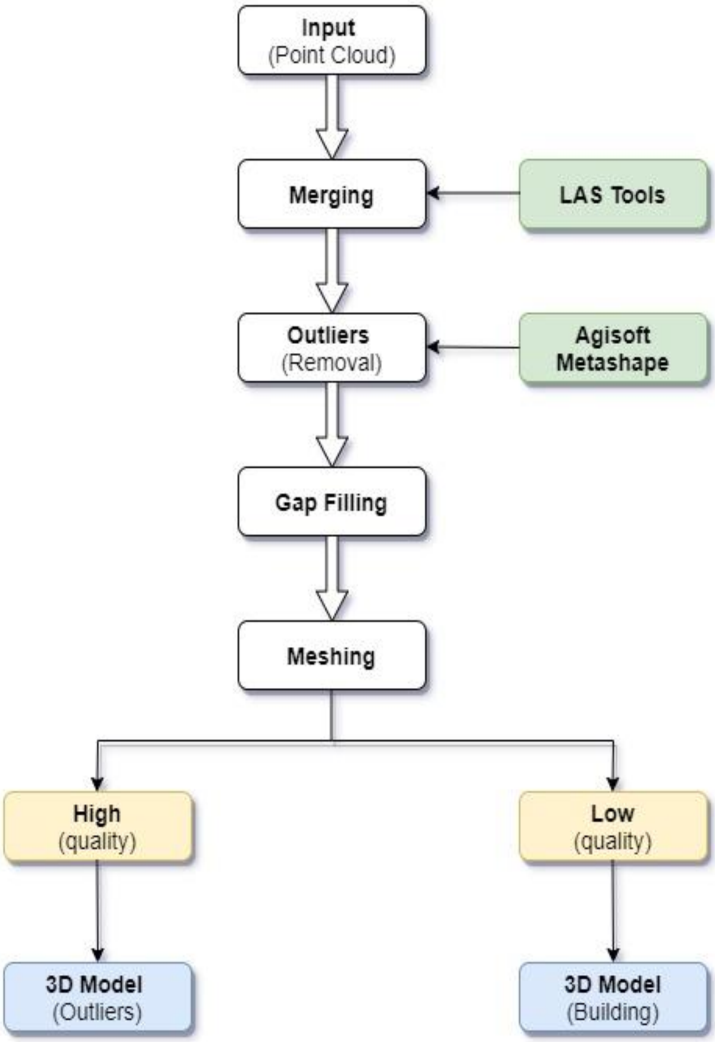
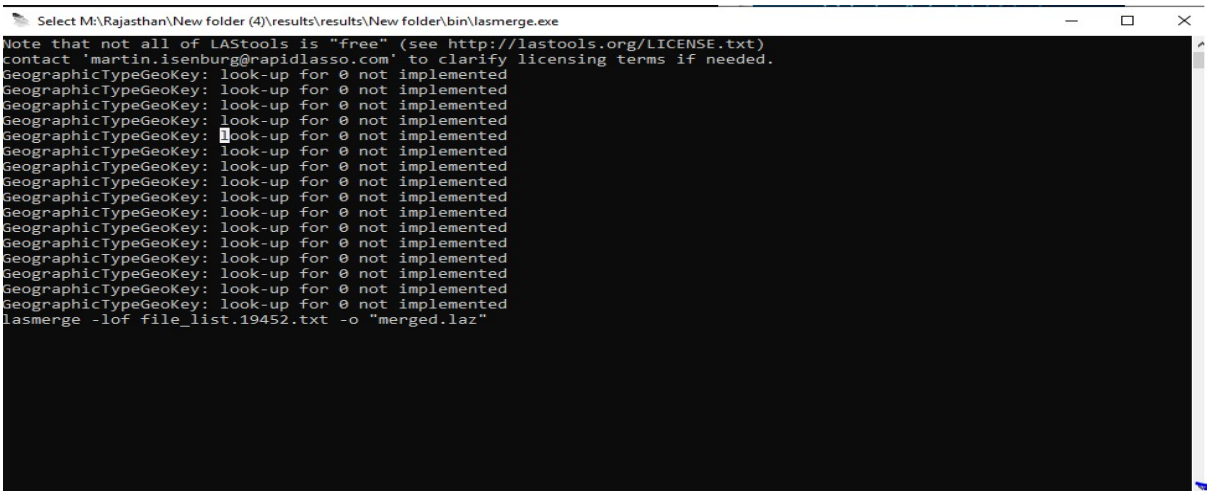


Figure 27: Automation Technique Methodology

6.2 Merging PCD Using LAS Tools

In this research, all of 16 scans are considered for the automation technique. The 16 raw scans with an overlap of 60% are converted to the software identifiable format and is imported into the LAS Tools [Sub-chapter 2.4.1.1] for merging the files as a single one. The LAS Tools uses LAS merge, which is a basic point cloud tool for merging the LAS file format. The 16 LAS files are merged as a single LAS file without any errors using scriptable command line codes. The below figure represents the successful implementation of all 16 LAS files, that are to be merged as a single file using LAS merge:


Select M:\Rajasthan\New folder (4)\results\results\New folder\bin\lasmerge.exe

```
Note that not all of LAStools is "free" (see http://lastools.org/LICENSE.txt)
contact 'martin.isenburg@rapidlasso.com' to clarify licensing terms if needed.
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
GeographicTypeGeoKey: look-up for 0 not implemented
lasmerge -lof file_list.19452.txt -o "merged.laz"
```

Figure 28: LAS Merge (16 LAS Files)

The merged LAS files are visualized in Agisoft Metashape [Sub-chapter 2.4.1.2] as a single file for inspection of the entire 3D point cloud data. The merged PCD is shown below:

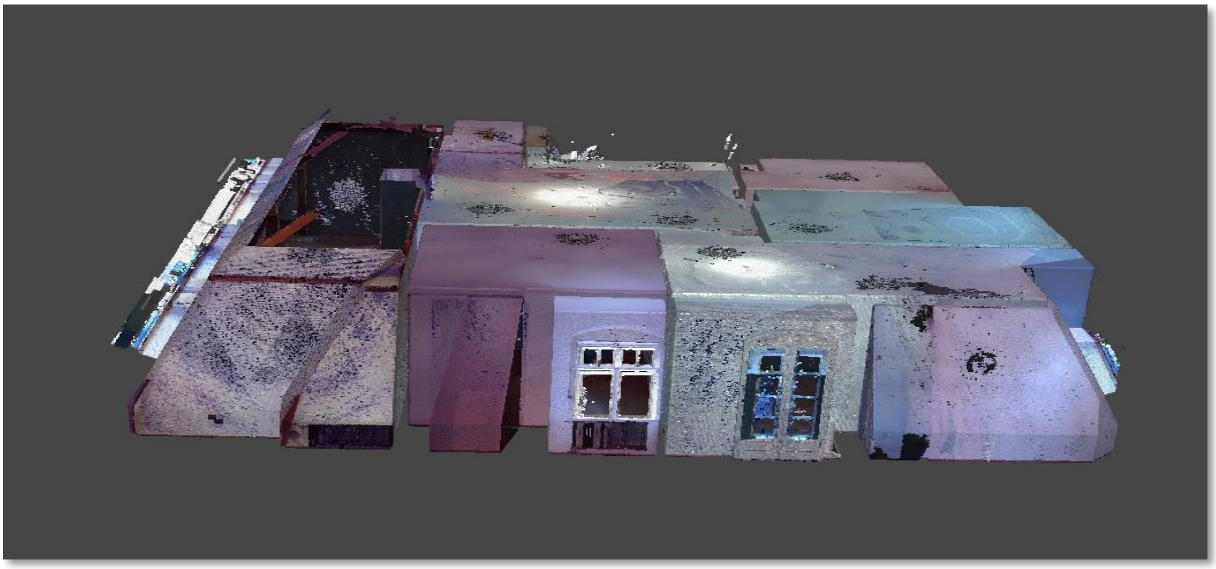
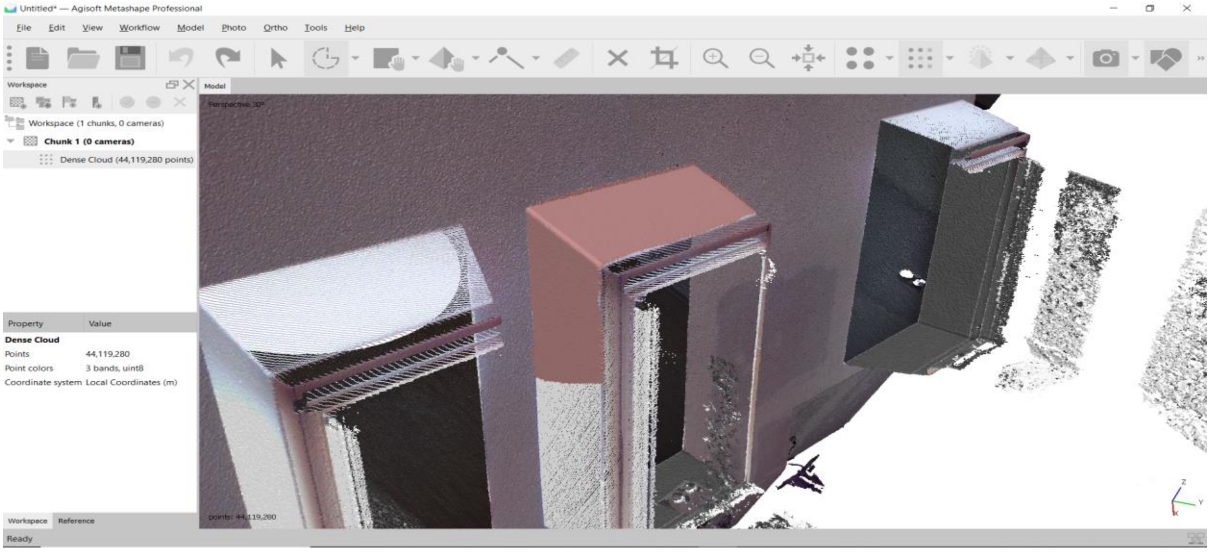


Figure 29: Merged PCD

6.3 Outlier Removal

The merged point cloud data is inputted into the Agisoft Metashape software and the outliers of the point cloud data are removed manually. The outliers such as supporting beams, windows, floor mat, heaters, working materials, toilet materials etc are removed manually using in-built software selection tools. The selection tool consists of rectangular, circular and free-form selection, which is used in different places for careful and error-free extraction of points. The below figure represents the selection of outliers from a point cloud data:



Property	Value
Dense Cloud	
Points	44,119,280
Point colors	3 bands, uint8
Coordinate system	Local Coordinates (m)

Figure 30: Outlier Selection (PCD)

The outliers are removed for the entire PCD using the suitable outlier selection tool and is visualized as a second layer. The outliers that are removed from the merged main point cloud data are shown below:

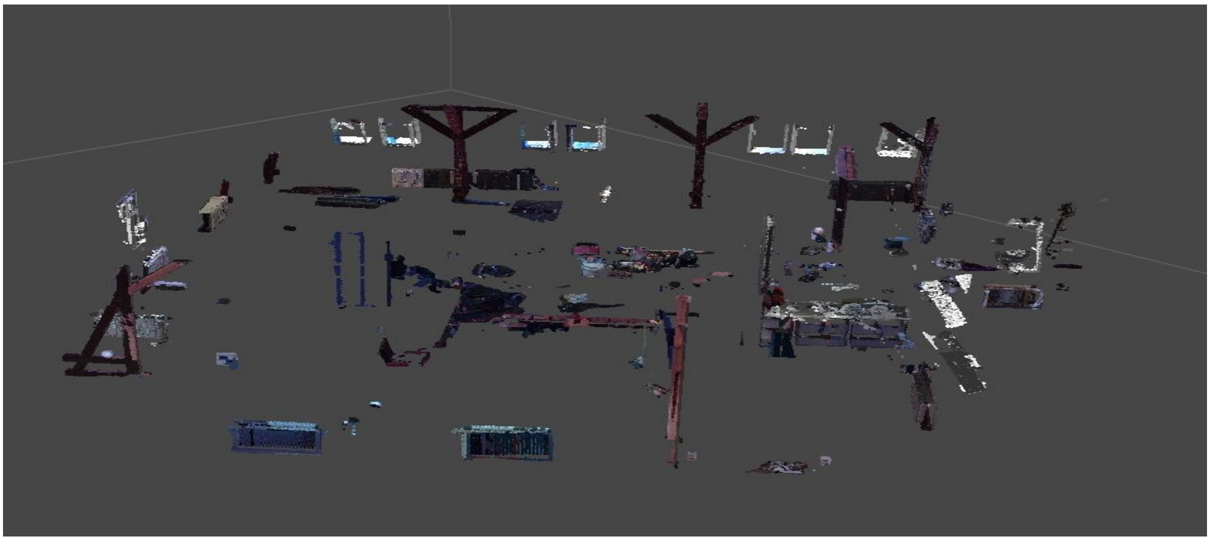


Figure 31: Outliers (PCD)

6.4 Purified PCD

After the removal of all outliers from a merged point cloud data, the indoor point cloud environment is purified with high dominant massive removal of noises/obstacle points. The purified point cloud data is taken and considered as a primary layer for this research, whereas the outliers of the point cloud data is considered as a secondary layer. In this research, the automation technique has shown an innovative idea and path, such that the combined use of powerful LIDAR point cloud software's with proper hardware specificity helps to easily solve the research objective and goal of the master thesis. The below figure represents the purified point cloud data:

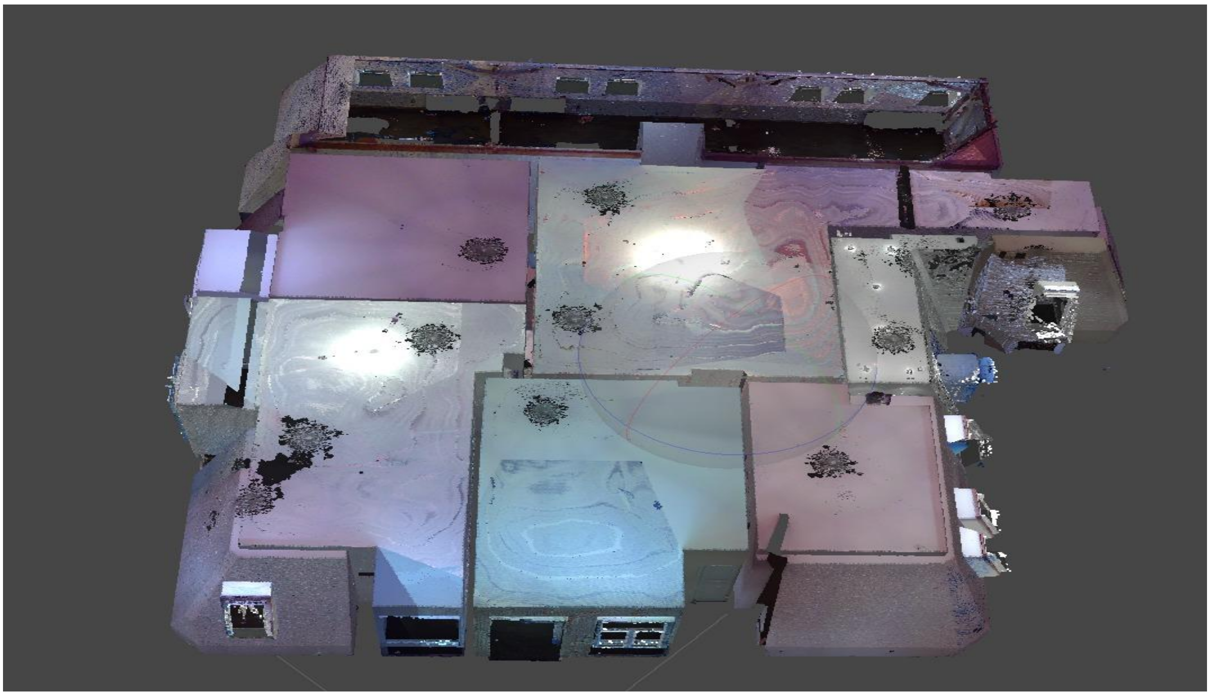


Figure 32: Purified PCD

6.5 Gap Filling

Due to high removal of outliers from the merged point cloud data and the gaps that occurred below the tripod on scanning needs to be interpolated and filled. Due to flat horizontal and vertical structures, tri-linear interpolation technique is carried out for the purified point cloud data. The trilinear interpolation is a multivariate interpolation technique, which interpolates the missing values or variables on three-dimensional axis. The trilinear interpolation technique can be applied to a dense regular gridded or scattered point cloud data. The interpolation technique interpolates the missing values by approximating the value of the function at an intermediate point within the local axis of the grid. Based on the neighbouring values the intermediate missing values are interpolated and filled. The below figure represents the ground surface that are interpolated using Trilinear interpolation:

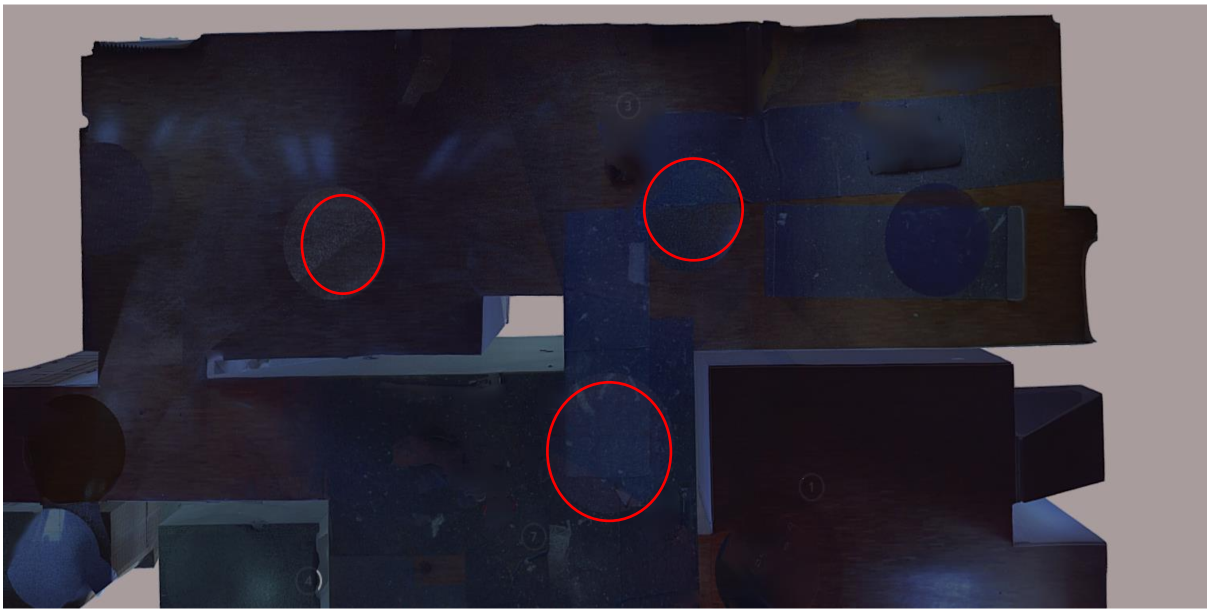


Figure 33: Interpolated PCD

6.6 Mesh Generation

In this research, the interpolated purified point cloud and outlier point cloud data are taken and considered for static mesh model. The mesh is generated by using powerful inbuilt machine learning algorithms, that are present in Agisoft Metashape software [Sub-chapter 2.4.1.2]. The algorithm uses Poisson surface reconstruction to triangulate the 3D points and the surface mesh is constructed and generated. The number of vertices and triangles created for both static models are tabulated below:

Static Mesh Model	No of Triangles	No of Vertices
Building	14.6 Million	7.3 Million
Outliers	7.1 Million	3.6 Million

Table 10: Model Geometry Specification

Due to high density of point cloud data, storage space and processing power only limited mesh qualities are considered for the static model. The purified indoor building point cloud data which has high density and number of 3D points, a low-quality mesh model is generated for visualization. The below figure represents the static mesh model of the purified building indoor point cloud data:

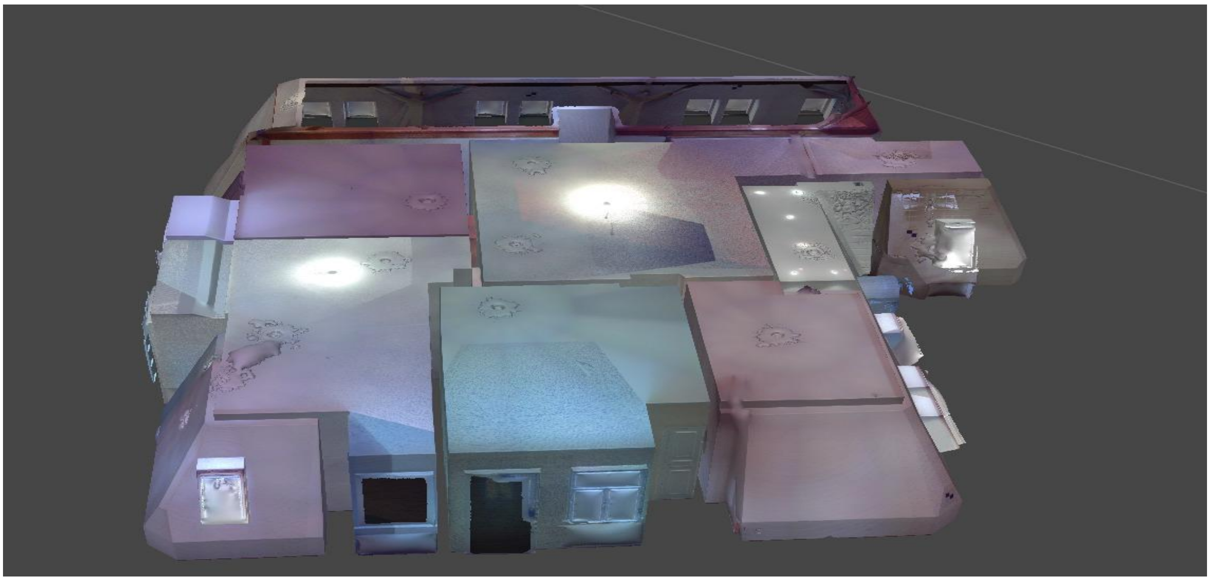


Figure 34: Building Model

The outlier point cloud data which has low density and number of 3D points, a high-quality mesh model is generated for visualization. The texture and colour of the static mesh models are preserved whereas the qualities between the two static models varies. The below figure represents the static mesh model of the outlier point cloud data:

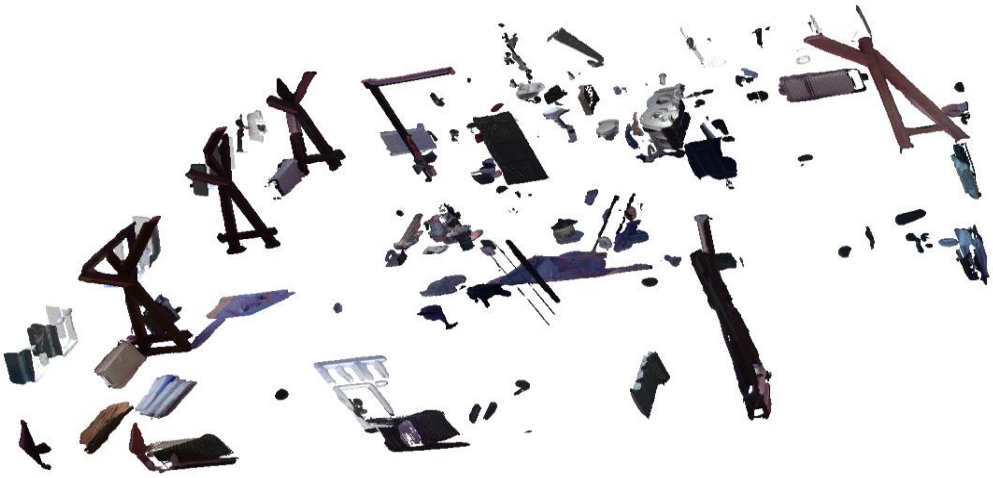


Figure 35: Outlier Model

RQ. 4 - What is the effort and certainty in interactively performing the automated task?

The Agisoft Metashape software is interactively used for performing the task to purify the model to its architectural shell. The use of automated technique provides more functionality and efficiency to interactively perform the task with very high effort and certainty. The combined use of LAS Tools, Metashape software and equivalent hardware specificity used in this research helps to perform the task in considerable and durable time with very highly efficient and successful outcome for the master thesis.

7 VISUALIZATION

For the research, visualization is the second important part that provides a technique for interactively communicating the 3D mesh models and point clouds using web-based tools and technologies [Sub-chapters 2.5.1.1 & 2.5.1.2]. This chapter describes briefly about the visualization of the point clouds and static mesh models, which is followed by basic and advanced visualization functionalities and operations to operate for the end user's in a web-based environment. The sub-chapters describe detailly about the web tools that are used for the purified point clouds and purified mesh models to interact and play with the point clouds and model's using higher-end functionalities and scientific visualization elements.

7.1 Point Cloud Data

The purified building indoor point cloud data and the outlier point cloud data are considered for primary visualization in this research. The opensource and web-based Potree Converter [Sub-chapter 2.5.1.2] is used for visualizing the point clouds in a web-based environment. The Converter uses Apache server for storing and accessing of the point cloud data. The Potree viewer helps to visualize the point cloud data on web with basic and advanced visualization elements. The visualization elements used for interactive operations on point clouds are tabulated below:

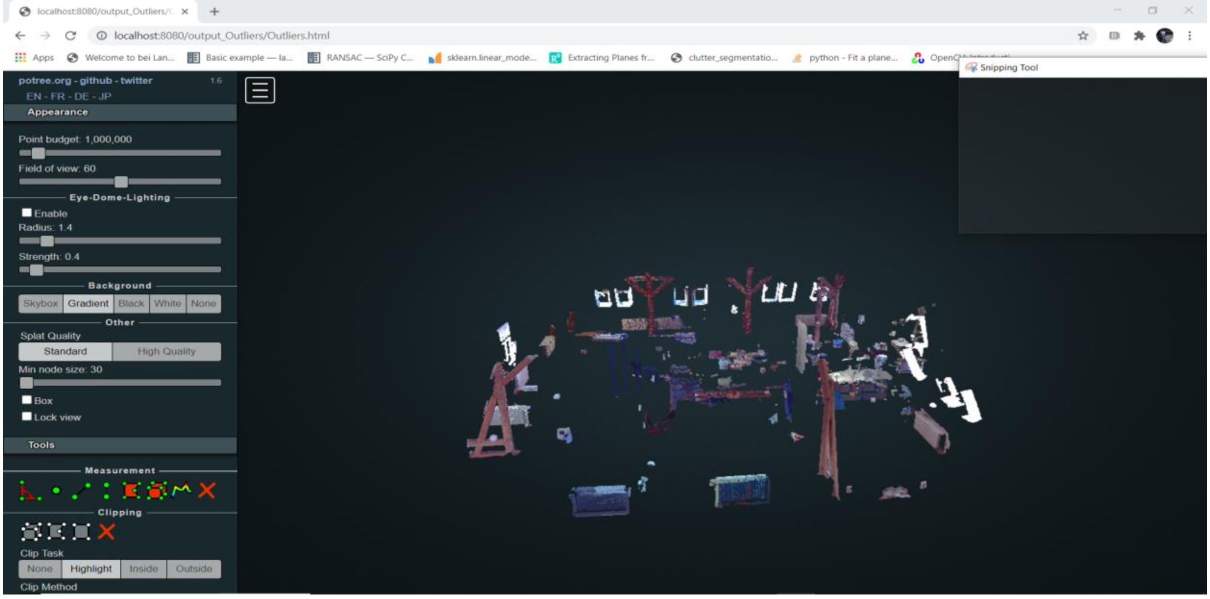
Visualization Elements	Description
Appearance	Provides functionalities on point cloud budget and field of view angle
Eye-Dome-Lighting	Provides functionalities on dome lighting effect on point clouds with radius and strength parameter
Background	Provides functionalities on background colour change
Splat Quality	Provides functionalities on point cloud quality and node size

Measurement Tools	Provides functionalities on point cloud measurement (point, line, height, volume and area)
Clipping Tools	Provides functionalities on clipping of point clouds (box clip, polygon clip and volume clip)
Navigation	Provides functionalities on point cloud navigation (earth control, fly control, orbit control etc)
Projection	Provides functionalities on different camera projection of point clouds (perspective & orthographic)
Scene Export	Provides functionalities on export of point clouds as a scene (JSON, DXF)
Classification Filter	Provides functionalities on visualizing the classified point cloud data (building, trees, forest, roads etc)

Table 11: Visualization Elements (PCD)

7.1.1 Outlier PCD Web Viewer

The Outlier point cloud data is inputted into the Potree converter and is converted to the compatible Potree viewer format. The converted format is stored in the Apache Server for accessing the point cloud data. The data is accessed by the Potree viewer from the Apache Server for visualization. The outlier point cloud data is visualized on web by defining the localhost and local server components. The below figure represents the visualization of outlier point cloud data on Potree web viewer:



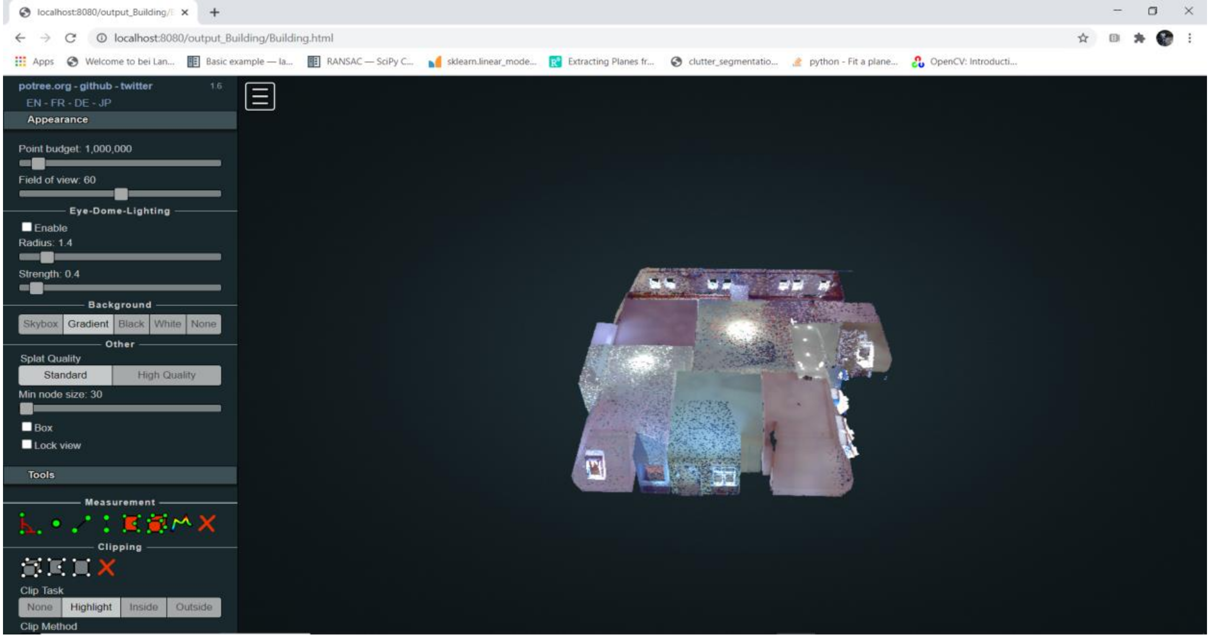
localhost:8080/output_Outliers/Outliers.html

Point budget: 1,000,000
Field of view: 60
Eye-Dome-Lighting
□ Enable
Radius: 1.4
Strength: 0.4
Background: Gradient
Splat Quality: Standard
Min node size: 30
□ Box
□ Lock view
Clip Task: Highlight

Figure 36: Outlier Web Viewer

7.1.2 Building PCD Web Viewer

The Building point cloud data is fed into the Potree Converter and the new indexes are built for each point and is converted to a compatible Potree format. The converted format is stored in the Apache Server and a local host (8080) is generated. The data is accessed by the Potree viewer and is visualized on web with high scientific visualization elements for interaction. The below figure represents the visualization of Building point cloud data on Potree web viewer:



localhost:8080/output_Building/Building.html

Point budget: 1,000,000

Field of view: 60

Eye-Dome-Lighting

□ Enable

Radius: 1.4

Strength: 0.4

Background: Gradient

Splat Quality: Standard

Min node size: 30

□ Box

□ Lock view

Clip Task: Highlight

Figure 37: Building Web Viewer

7.2 Point Cloud Mesh Model

The purified building mesh model and outlier model are considered for secondary visualization in this research. The opensource and web-based Sketchfab tool [Sub-chapter 2.5.1.1] is used for visualizing the mesh models in a web-based environment. The building and outlier mesh models are visualized and can be interacted with various visualization elements and functionalities on web. The models are published on internet and can be viewed, interacted and downloaded for free. The basic and advanced scientific visualization elements used for interactive operations on mesh models are tabulated below:

Visualization Elements	Description
Navigation	Provides functionalities on model navigation (orbit view & first-person view)

Model Inspector	Provides functionalities to inspect the models (geometry, Material channels, render, viewport & object colour)
Annotations	Provides functionalities to annotate the mesh models using numbering technique
Virtual Reality	Provides functionalities to visualize the model using Virtual Reality technology
Help	Provides functionalities on control options help settings

Table 12: Visualization Elements (Mesh Models)

7.2.1 Outlier Web Model

The high-quality outlier mesh model created from an outlier point cloud is imported into the Sketchfab account and the basic 3D properties are edited for the model. The basic 3D properties consist of projection, angle rotation, colour intensity, background colour, field of view, rendering options, adding videos and music etc for the model. The preliminary properties are edited, saved and published on internet using Sketchfab account. The link to access the outlier model on web are:

https://skfb.ly/6TAwr

The below figure represents the outlier mesh model, which is annotated based on numbering and the visualization elements that are present for the end user to operate are shown below:

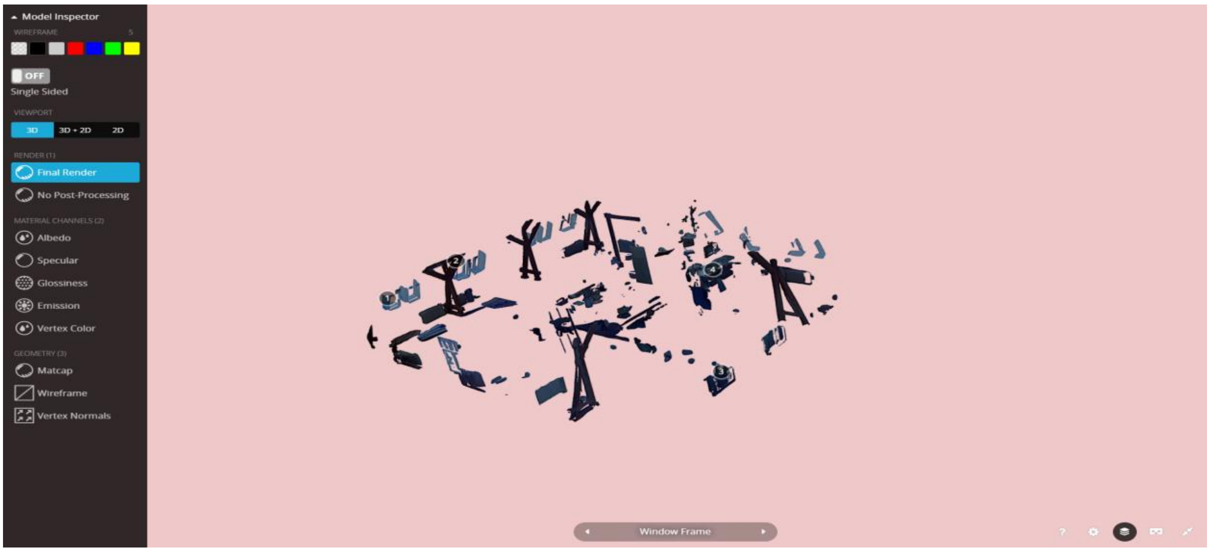


Figure 38: Outlier Web Model Viewer

7.2.2 Building Web Model

The purified building mesh model created from the purified point cloud data is imported into the Sketchfab account and the basic and advanced 3D properties of the model are edited. The pre-defined 3D properties are edited on web and is stored, published and visualized on web using Sketchfab account. A web-link access is provided for the published model on internet. The link to access the published building web model are:

https://skfb.ly/6TCqT

The below figure represents the building mesh model, which is annotated based on numbering technique and displays the various departments of the office building. The annotated mesh model with high visualization functionalities for the end user to operate are shown below:

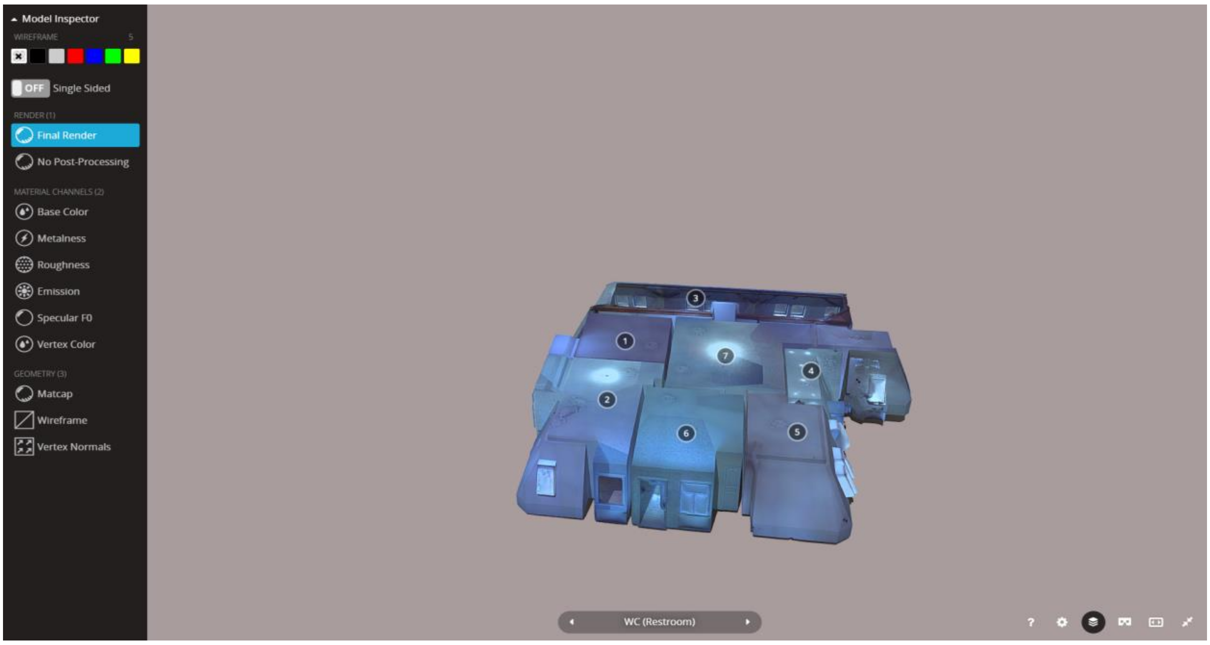


Figure 39: Building Web Model Viewer

8 RESULTS & DISCUSSION

8.1 Detection of Purified Room Geometry Using Computer Vision

The purified indoor room geometry is detected using Ransac computer vision algorithm and the planes are predicted for two test samples [Chapter 4]. Based on the defined parameters and consideration of high favourable iterations of planes, the suitable corresponding points are predicted and segmented from outliers and obstacles. The python code for the application of sample models are given in Appendices. Due to limitation of the defined Ransac algorithm [Sub-chapter 4.4], only inclined planes and horizontal planes are predicted and visualized. The below figure represents the detection of purified point cloud room geometry using Ransac computer vision algorithm:

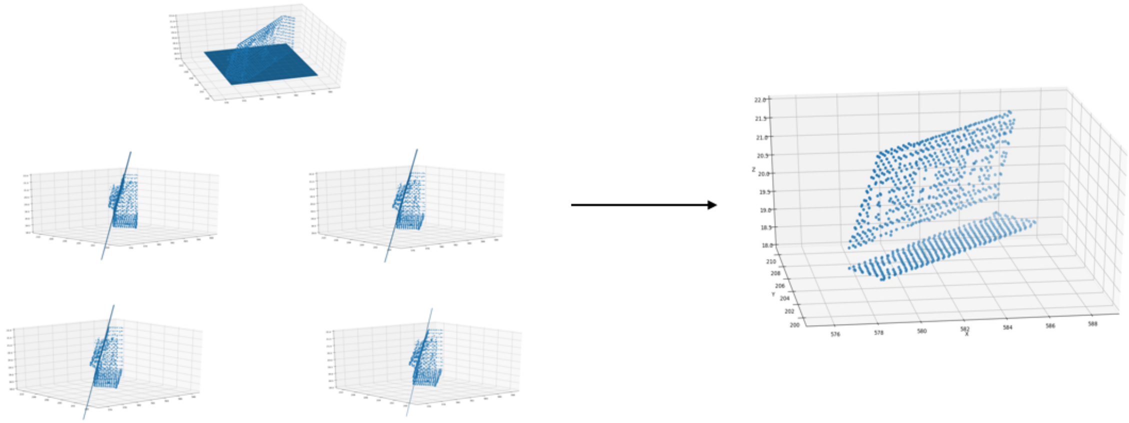


Figure 40: Purified Room Geometry (Sample 1)

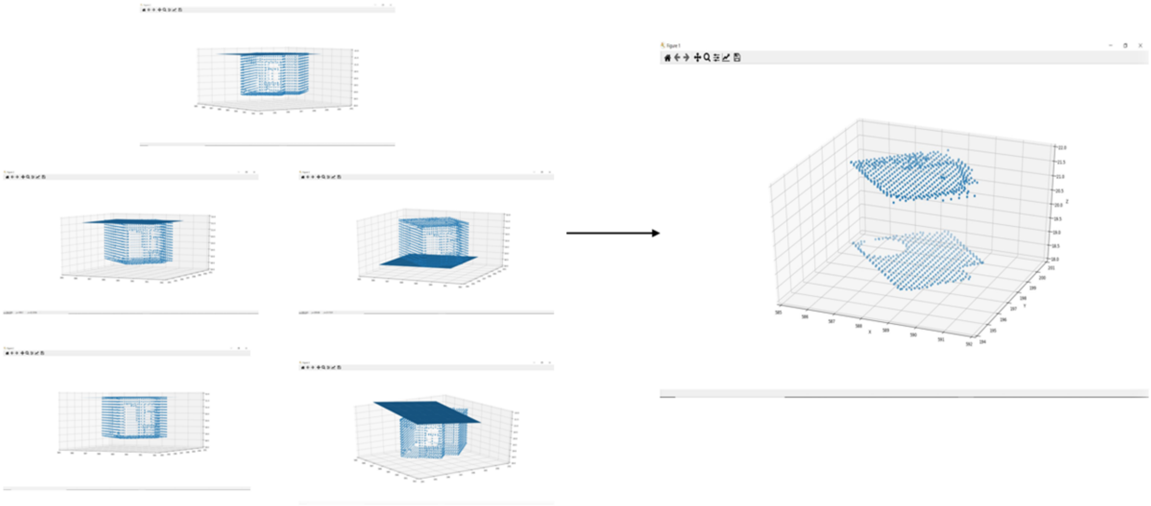


Figure 41: Purified Room Geometry (Sample 2)

From the above research [Chapter 4], the powerful and robust computer vision algorithm enables to predict the purified indoor point cloud geometry with high level suppression of obstacles and noises, based on the defined supporting parameters. The parameters of the algorithm are defined in a very clear state, such that only suitable planes and points are predicted to create a purified room geometry (wall, floor and ceiling). The high suitable consideration of planes leads to the high-level prediction of the room geometries. Thus, the combined use of computer vision algorithm with LIDAR technique has provided an innovative idea and path for solving the research problems as well as the real-world problems of the master thesis.

8.2 Detection of Purified Room Geometry Using Automation Technique

The purified indoor point cloud room geometry is detected using automated software-based approach [Chapter 6]. The powerful inbuilt machine-learning and computer visions algorithms helps to detect the purified room geometry in a very clear and precise way. The proper and combined selection of software's and tools lead to the high precision of valid outcome and correctness. The below figure represents the detection of purified room geometry using automation technique:

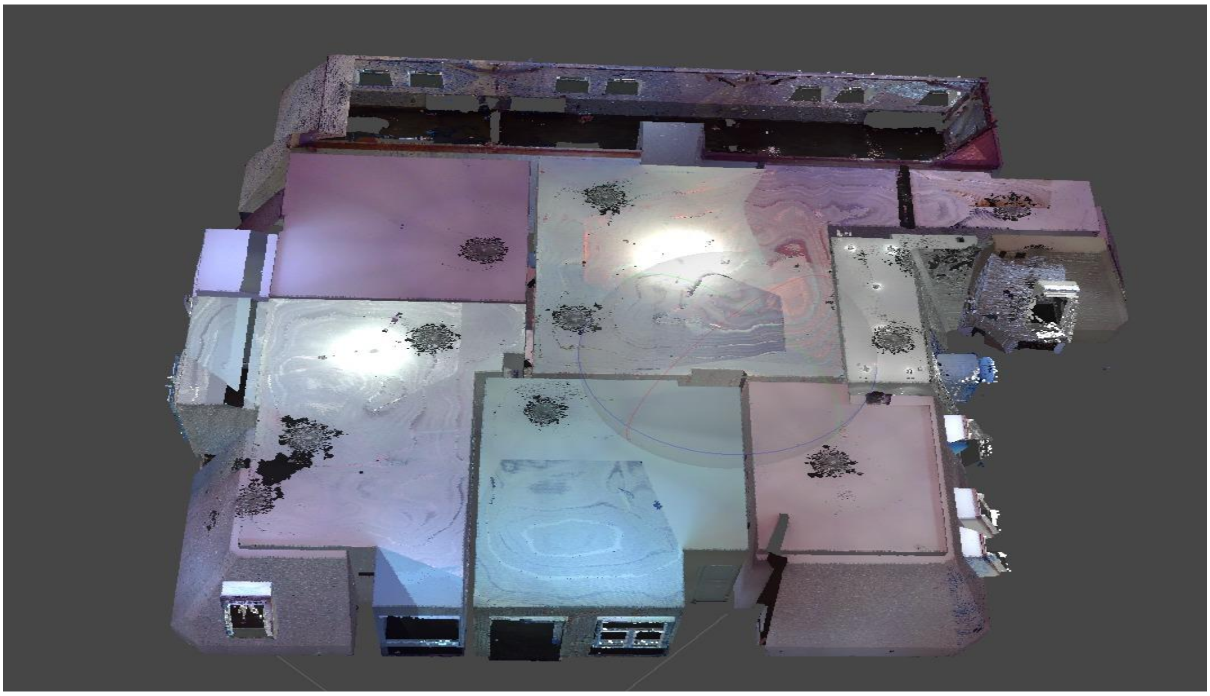


Figure 42: Purified Room Geometry (PCD)

From the above research [Chapter 6], the inbuilt software algorithms with proper plugins and tools enables to select and remove the outliers/obstacles from a merged point cloud data to its architectural shell. The purified room geometry is inspected and converted to a model for interactive visualization using web tools and technologies. Thus, the proper valid and combined selection of software's and hardware components for point cloud data has provided a new era for solving the real-world problems and research objectives in a very easy and an efficient way for the master thesis.

9 CONCLUSION & FUTURE WORK

From the above research, the application use of both semi-automation and automation technique on point clouds has shown an innovative and ingenious idea for solving the research questions and objectives of the master thesis in a very easy and highly efficient way. The research questions (RQ) are answered spontaneously at the end of each corresponding chapters by solving the research objectives. The use of defined computer vision algorithm with point cloud data (limitations) provides less accuracy and correctness when compared to the inbuilt software algorithms. But this is not the case for all computer vision algorithms that are defined for solving the real-world research objectives and problems. Based on the defined criteria for inclined and horizontal planes, the use of program-based approach has shown higher proximity of completeness and correctness for detecting the room geometry planes when compared to the in-built Blackbox algorithm present in the automated software. Due to limitations of the defined algorithm, the upper hand goes to the automated software for this research. Based on the limitations of the algorithm in the semi-automation technique, the automated software-based approach is considered for modelling and visualization process. The end user product is a higher end purified model with interactive visualization elements, which is mainly created to focus on engineers, planners and architects to play, visualize, analyse and for future renovation of the building model.

Due to limitations of the defined Ransac algorithm, the semi-automation technique needs to be improved for highly efficient outcome and accuracy perspective. The future works that are need to be carried out for the semi-automation technique are the prediction of vertical wall planes and segmentation of points from the predicted wall plane. The Ransac algorithm needs to be improved by defining loop concepts, automatic point extraction and automatic processing of point clouds using Deep learning Neural Networks for higher accuracy and precision of the output products. The main future work involves the upgradation of the defined Ransac algorithm with more supporting libraries, external plugins and the use of Deep Neural Network concepts.

The above-mentioned future works needs to be carried out for better outcome and result perspective, so that the semi-automation technique can gain an upper hand over inbuilt software algorithms. Thus, the application use of both automation and semi-automation technique provides various insight and knowledge for solving the research objectives and goals of the master thesis.

References

[1] Li, L., Yang, F., Zhu, H., Li, D., Li, Y., & Tang, L. (2017). An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells. Remote Sensing, 9(5), 433. doi: 10.3390/rs9050433.

[2] Bool, D. L., Mabaquiao, L. C., Tupas, M. E., & Fabila, J. L. (2018). Automated Building Detection Using Ransac From Classified Lidar Point Cloud Data. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W9, 115–121. doi: 10.5194/isprs-archives-xlii-4-w9-115-2018.

[3] Kurban, R., Skuka, F., & Bozpolat, H. (2015). Plane Segmentation of Kinect Point Clouds using RANSAC. The 7th International Conference on Information Technology. doi: 10.15849/icit.2015.0098.

[4] Yang, M. Y., & Förstner, W. (2010). Plane Detection in Point Cloud Data. (pp. 1-16). (IGG : Technical Report ; Vol. 1, 2010). Bonn: University of Bonn.

[5] Lan, J., Tian, Y., Song, W., Fong, S., & Su, Z. (2018). A Fast Planner Detection Method in LiDAR Point Clouds Using GPU-based RANSAC. UMCit@KDD.

[6] Zeineldin, R.A., & El-Fishawy, N.A. (2016). Fast and accurate ground plane detection for the visually impaired from 3D organized point clouds. 2016 SAI Computing Conference (SAI), 373-379.

[7] Ransac algorithm (2D). Slides from University of Illinois Lectures by D. Hoiem. First Accessed: 2020-05-15.

[8] LAS Tools. https://www. https://rapidlasso.com/lastools/. First Accessed: 2020-06-03.

[9] Agisoft Metashape Software. https://www.agisoft.com/. First Accessed: 2020-06-05.

[10] Sketchfab. https://sketchfab.com/. First Accessed: 2020-07-15.

[11] Potree Converter. http://www.potree.org/getting_started.html. First Accessed: 2020-07-30.

[12] PLY File Format. https://en.wikipedia.org/wiki/PLY_(file_format)#:~:text=PLY%20is%20a%20computer%20file,list%20of%20nominally%20flat%20polygons. First Accessed: 2020-04-15.

[13] LAS File Format. https://en.wikipedia.org/wiki/LAS_file_format#:~:text=The%20LAS%20(LASer)%20format%20is,industry%20standard%20for%20lidar%20data. First Accessed: 2020-06-01.

[14] Study Area (FARO 3D Software GmbH). Google Earth Image. https://www.google.com/maps/. First Accessed: 2020-04-01.

[15] Methodology Graphs. https://www.draw.io/. First Accessed: 2020-05-01.

[16] CloudCompare. Ransac Shape Detection Plugin. https://www.cloudcompare.org/doc/wiki/index.php?title=RANSAC_Shape_Detection_(plugin). First Accessed: 2020-05-27.

[17] Metadata of 16 Raw Scans. First Accessed: 2020-04-15.

APPENDICES

Appendix A: Python Code for Visualization and Down-sampling of PCD

For visualization and down-sampling of point cloud data in chapter 4 [4.2, 4.2.1, 4.3, 4.3.1]:

```
# Visualization and Down-sampling of PCD

import numpy as np

import open3d as o3d

pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/Sample1&2.ply")

o3d.visualization.draw_geometries([pcd])

downpcd = pcd.voxel_down_sample(voxel_size=0.2)

xyz = np.asarray(downpcd.points)

o3d.visualization.draw_geometries([downpcd])

o3d.io.write_point_cloud("F:/Faro Software/Output file (X,Y,Z)/xyz_1.ply", downpcd)

np.savetxt("F:/Faro Software/Output file (X,Y,Z)/xyz_1mainvalue.csv", xyz, delimiter=",")
```

Appendix B: Python code for Ransac Plane Detection and Point Prediction

For Ransac plane detection and point prediction of point cloud samples in chapter 4 [4.2.2, 4.2.3, 4.3.2, 4.3.3]:

```
import numpy as np
import math
from sklearn import linear_model
import open3d as o3d
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
#Defining and Finding a Plane
def find_plane(pcd):
    xyz = np.asarray(pcd.points)
    np.savetxt("F:/Faro Software/Output file (X,Y,Z)/xyz.csv", xyz, delimiter=",")
    XY = xyz[:, :2]
    print(XY)
    Z = xyz[:, 2]
    print (Z)
    #X = xyz[:, 0]
    #YZ = xyz[:, 1:]
    #print(xyz)
ransac = linear_model.RANSACRegressor(linear_model.LinearRegression(),residual_threshold=0.01, max_trials=1,stop_probability=0.99)
ransac.fit(XY, Z)
#ransac.fit(YZ, X)
a, b = ransac.estimator_.coef_
d = ransac.estimator_.intercept_
p = ransac.get_params(True)
print(p)
o = ransac.predict(XY[:,:2])
print(o)
```

```
np.savetxt("D:/Project Outputs/Part_1/Python/Box Room/Iterations/Itr 20/zvalue.csv", o,
delimiter=",")
new = np.asarray(o)
print(new)
plt.plot(new)
print(a, b, d)
return a, b, d
#Defining the plane equation with placing X axis
def angle_rotate(a, b, d):
    x = np.arange(10)
    y = np.arange(10)
    X, Y = np.meshgrid(x, y)
    Z = a * X + b * Y + d
    rad = math.atan2(Y[1][0] - Y[0][0], (Z[1][0] - Z[0][0]))
    return rad - math.pi
def show_graph(X, Y, Z):
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    ax.plot_surface(X, Y, Z)
    plt.show()
#Finding Plane Points
if __name__ == '__main__':
    pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/xyz_1.ply")
import ransac
#Visualization of the point cloud
pcd = o3d.io.read_point_cloud("F:/Faro Software/Output file (X,Y,Z)/xyz_1.ply")
o3d.visualization.draw_geometries([pcd])
a, b, d = ransac.find_plane(pcd)
print(ransac.angle_rotate(a, b, d))
```

```
#downpcd = pcd.voxel_down_sample(voxel_size=0.0001)

#o3d.io.write_point_cloud("F:/Faro Software/Output file (X,Y,Z)/xyz.ply", downpcd)

xyz = np.asarray(pcd.points)

#np.savetxt("D:/Project Outputs/Part_1/Python/Box Room/Iterations/Itr 20/xyzvalue.csv", xyz, delimiter=",")

min_x = np.amin(xyz[:, 0])

max_x = np.amax(xyz[:, 0])

min_y = np.amin(xyz[:, 1])

max_y = np.amax(xyz[:, 1])

x = np.arange(min_x, max_x)

y = np.arange(min_y, max_y)

X,Y = np.meshgrid(x,y)

Z = a*X + b*Y + d

fig = plt.figure()

ax = fig.gca(projection='3d')

ax.set_xlim3d(585,592)

ax.set_ylim3d(194,201)

ax.set_zlim3d(18,22)

ax.plot_surface(X,Y,Z)

ax.scatter(xyz[:, 0], xyz[:, 1], xyz[:, 2])

plt.show()
```

Appendix C: Python code for Plotting the Predicted Purified Indoor Points

For plotting the predicted purified indoor points for samples in Chapter 4 [4.2.5, 4.3.5]:

```
# Visualization of Predicted Purified Indoor Points

import numpy as np

import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

data = np.genfromtxt("D:/Project Outputs/Part_1/Python/Box Room/Output Points/newzvalues.txt")

fig = plt.figure()

ax = fig.gca(projection='3d')

ax.set_xlim3d(585,592)

ax.set_ylim3d(194,201)

ax.set_zlim3d(18,22)

ax.scatter(data[:, 0], data[:, 1], data[:, 2])

ax.set_xlabel('X')

ax.set_ylabel('Y')

ax.set_zlabel('Z')

plt.show()
```