DEPARTMENT OF DEVELOPMENT AND PLANNING

AALBORG UNIVERSITY
DENMARK

Semiautomatic Building Information Modelling from point clouds

Specialization in Surveying and Mapping

4th Semester MSC in Surveying, Planning and Land Management

Călușer Andrei
&
Cristian George

- Aalborg, June 2018 -

Title:

Semiautomatic Building Information Modelling from point clouds

Project:

Master's Thesis

Period:

February 2018 – June 2018

Project Group:

Călușer Andrei
Cristian George

Supervisors:

Karsten Jensen

Number of pages:

76

Number of printings:

0 (none)

Number of appendices:

3

Abstract

The goal of this project is to explore the possibility of an automatic model creation from 3D data for later integration with BIM.

In the project, BIM is defined in its current state as well as what point clouds are and some methods to acquire them. A data collection method is presented for the model creation, followed by a brief description of some of the methods currently in use for this purpose.

Later, an attempt at creating a model is made to be exported to a CAD program and its accuracy assessed.

Acknowledgements

The following project was written at Aalborg University, Denmark. It is the last (4th) semester's project and it constitutes the master's thesis. Here we would like to tank the people that helped in making this project possible and whose input was greatly appreciated.

First and foremost, Karsten Jensen, our supervisor. His guidance, support and advice (both academic and otherwise) throughout the conception of the project made it a comfortable and engaging environment to work in, for which we are truly grateful. Also, for establishing the initial connection with COWI.

We would also like to thank COWI, one of Denmark's largest civil engineering firm, for providing the instrumentation to collect the data. Thanks to Niels Houen, surveyor and regional manager at COWI, who dedicated his time to teach us how to properly use the scanning instruments and the insightful thoughts on data collection quality.

Contents

Introduction

BIM stands for Building Information Modelling. The exact definition of the term varies slightly from discipline to discipline, but, we can say that this is how buildings and other large structures are modeled today. The US National Building Information Model Standard Project Committee has the following definition:

Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition. [1]

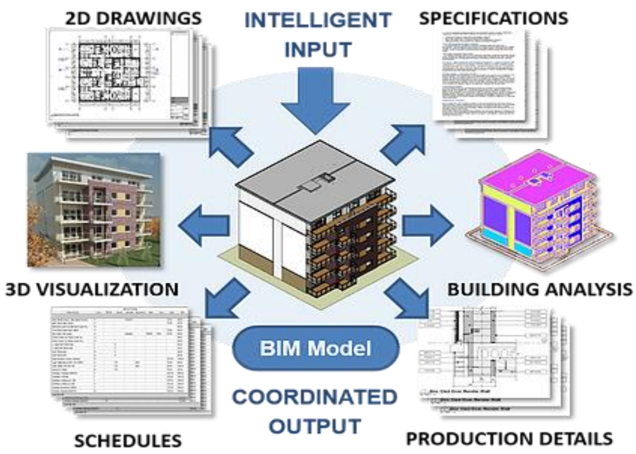


Figure 1 Principal components of a BIM [2]

Usually, BIM models are based on 3D models, also including other layers of information, such as: time, costs, materials, relationships between objects etc. The additional information that BIM brings to the projects can be useful when performing all kinds of advanced analyses of the model. Based on the extra information layers from a BIM, light analysis, geographic information, quantities and properties of building components can be made without additional visits to the building site.

Unfortunately, the point cloud to BIM process is largely manual, due to large amounts of data and various problems that raise in the model reconstruction. Thus, requiring a good set of skills and huge amounts of time. A main challenge today is the automation of BIM processes from point clouds. The aim of the current project is to develop a workflow that will allow the user to extract as much information as possible from the point cloud to create an as complete as possible BIM and integrate it with an CAD software. In the following chapters we will further explore what a BIM is and how can it be created, as well as expand on the importance and uses of BIM and the three main steps of creating one: modelling the geometry of components, assigning an object category and material properties, and establishing of relationships between components.

Initial problem statement

The creation of Building Information Models became a well establish practice for new buildings, but, the majority of existing buildings do not have an information model yet. The BIM implementation in existing buildings has a multitude of potential benefits such as: restoration, documentation, maintenance, energy and space management. The main issue is that the process of transforming data from the survey to BIM is a mostly manual due to the large amount of data. The main challenge today is to automate the process leading from point clouds to an as-build BIM.

The goal of this project is to develop a semiautomatic processing workflow which will allow the user to extract as much information as possible from the input data and to easily integrate the result in CAD software.

Preliminary questions

What is BIM?

1. Why create a BIM?
2. What are the advantages of point clouds in creation of BIMs?
3. How can a point cloud be created?
4. How can a BIM be created from a point cloud?

Acronym list

A list of acronyms used in this project.

CAD - Computed Aided Design

CAE - Computed Aided Engineering

BIM - Building Information Modelling

AEC – Architecture, Engineering and Construction

LiDAR – Light Detection And Ranging

RANSAC - RANdom SAmple Consensus

MLESAC – Maximum Likelihood Sample Consensus

RMSAC - Randomized M-estimator SAmple Consensus

LS – Least Squares

PCA – Principal Component Analysis

PCD – Point Cloud Data

ALS – Aerial Laser Scanning

TLS – Terrestrial Laser Scanning

Chapter 1 - Building Information Modelling

1. What is Building Information Modelling

From the second half of the 20th century, the use of traditional drawing and calculation tools stopped in favor of current digital systems. Currently, the most used tools among specialized personnel CAD/CAE systems. However, the next step is called BIM and it's getting a strong front in usage and adoption.

1.1 Definitions

The acronym "BIM" can be translated in two ways: Building Information Modelling or Building Information Management. In the course of this project the reference to the term BIM will be made from a "Building Information Modelling" perspective. Some of the definitions will be presented next, starting with a look at the industry and then explore some of the state/institution standard definitions.

Autodesk – maker of Revit and AutoCAD Civil 3D, BIM-enabled platforms

"BIM is an intelligent model-based process that helps make design, engineering, project and operational information accurate, accessible and actionable for buildings and infrastructure." Bond, PR manager for AEC and infrastructure at Autodesk. [5]

The definition for BIM from Autodesk's perspective is an information-centered one, high levels of data in the model help with cooperation between different parties that aid in the project.

Graphisoft – maker of ArhiCAD, one of the first BIM software platforms

"BIM is the use of 3D virtual models of buildings, as well as a process of managing and collecting building data," according to Moscarello (media relations consultant at Graphisoft). "When it comes to BIM, everything starts with a 3D digital model of the building."

It can be seen that Graphisoft has a model centered perspective. The model sets the base for the overall project, contains all the object elements and highlights the practical uses of those.

Bentley Systems - develops, manufactures, licenses, sells and supports computer software and services for the design, construction, and operation of infrastructure.

"Using a BIM methodology improves collaboration and ensures a new level of control over projects of all sizes," Harry Vitelli (senior VP of construction and field project delivery at Bentley Systems). "Better project outcomes are achieved through a complete flow of information among applications and across distributed project teams for greater accuracy across the entire supply chain."

In their definition, Bentley, discusses the aspects of information and modelling with emphasis on the process involved in BIM. This process, in concordance with technological aspects of BIM enable smooth cooperation between teams. [5]

In addition to the definition presented in “Introduction”, there are some other definitions made by state/institutions regarding BIM.

U.S. Government General Services Administration defines BIM as “the development and use of a multi-faceted computer software data model to not only document a building design, but to simulate the construction and operation of a new capital facility or a modernized facility." [6]

Another document issued by British Standard Institution Specification for information management for the capital/delivery phase of construction projects using building information modelling defines BIM as "the process of design, construction and use of the building or facility infrastructure using information about virtual objects." [7]

It can be seen that every source has a slight definition of what is BIM and what are its components, although all of them agree that BIM is both a best-practice process and a type of software application. By using BIM, the creation of a shared building project with integrated information is possible, which models both the structure and the entire timeline of the project from inception. The advantages of such a model will be presented further in the following sections.

1.2 Importance of BIM

Building Information models truly shine when it comes to coordinated projects. BIMs are a single piece of information source where all ideas, documents, functional characteristics, operational details, etc. of a construction project are kept, without losing or even creating any redundant data.

One of the most important attribute of a BIM, is that it captures reality. With the advance in data collection technology and better mapping tools, projects start to include aerial imagery, digital elevation and laser scans of existing infrastructure. With BIM everyone working on the project can benefit from all that input data compiled and shared within a model.

With a shared model, there is no need for rework or duplication of models for the different requirements of building disciplines. The model contains more information than a drawing, allowing each discipline to contribute to the final model with its own specific details. BIM drawing tools also have the advantage of being faster than 2D drawing tools. Besides being faster, each object from the model is connected to a database which is updated automatically as the model evolves. The computerized counting of elements alone, has been a significant labor and money saver.

Another important aspect of Building information models is that it improves collaboration. Sharing models is easier than sharing drawing sets, because of built in functions that are supported by a digital workflow. Most of the project management functionality can be delivered in the cloud (for example: Autodesk BIM 360 solutions). Here, every discipline has the tools to share their complex projects and to coordinate with their peers. With a model and an accurate set of sub-models for each phase of construction it is handy for all involved parties to create a more efficient construction process, delivering a predictable path to the expected outcome.

The BIM toolset also allow users to simulate and visualize different aspects of the building such as sunlight during different seasons or it can help automate clash detection of elements. By modelling first, these errors are discovered early, and costly on-site clashes are reduced. The model also assures perfect fit of elements which allows them to be manufactured elsewhere rather than on-site.

Last but not least, BIMs have the added benefit of being portable. With all the information being uploaded to the cloud, the user can access the model and the project details anywhere on any device.

1.3 What are the advantages of point clouds in creation of BIMs?

Further we will explore some of the advantages of having a point cloud as root in the creation of BIMs.

Point cloud data is immensely powerful for analysis on its own, however for the purpose of this project we will assume that the point cloud will be converted to object based BIM models.

1.3.1 3D Modelling

The most obvious advantage of creating a BIM from a point cloud is that by combining the project of the building with real terrain and surroundings from scanning serves for creation of realistic 3D models.

Another important factor is to decide what is the most appropriate representation of the building model. In some cases, the best representation for a point cloud would be a triangulated surface. This is mostly used for free-form objects where the original point cloud is altered (triangulated, smoothed and simplified). However, an object wise representation is more suitable than a huge triangulated surface in the context of modelling building. The desired model should be able to show different levels of detail depending on the user's needs.

1.3.2 4D BIM Scheduling

Having an accurate 3D representation of the scanned elements allows for further use of the data when it comes to the 4D (time) aspect associated with each element of the building. By combining scanning and scheduling improvements can be made when it comes to the renovation and maintenance of the building.

1.3.3 5D BIM Quantity and Cost

Another important information that can be obtained from a good 3D model is cost planning or 5D BIM. Point clouds produce 3D models that allow an accurate delineation of cost assemblies associated with new and existing work. Scanning and modeling the building before the execution allows for the cost buffers to be tied to the actual quantity of work resulting in a less dramatic impact on the overall estimate.

1.3.4 6D BIM Facilities Model

Last but not least, a clear benefit to using point clouds is when considering the deliverables that will be handed to the owner at the end of the project. Owners are very interested in having as much detail as possible about the as-built condition of the building. Scanning can be done at various stages of the building therefore the final position of the elements can then be cross-checked with the BIM to make sure that the handover model corresponds with reality.

In conclusion point clouds can be used to optimize the construction process while reducing the risk of the project, the cost, and the time required for its completion.

1.4 How can a point cloud be created?

A point cloud is a set of points within a 3D coordinate system. The most common way of obtaining a point cloud is through laser scanning. It can also be built using a total station, but it is a tedious task. Another option would be photogrammetry, which is more effective but unfortunately it is not usable in every situation.

In modern engineering the term “laser scanning” has two related but separate meanings. The first, is more general referring to the controlled deflection of laser beams, visible or invisible. This is used in stereolithography machines, laser engraving machines, laser printers, laser shows and barcode scanners.

The second meaning is more specific, referring to the controlled steering of laser beams followed by distance measurement at every pointing direction. This is called 3D object scanning or 3D laser scanning, and it is used to rapidly capture shapes of objects, buildings and landscapes.

Laser scanning began in the 60’s, but was mostly rudimentary. The early scanners used lights, cameras and projectors in an attempt to accurately recreate the surfaces of various objects and places. Due to limitations in technology it often took a lot of time and effort to perform this task. After 1985 they were replaced with scanners that could use white light, lasers and shadowing to capture a given surface.

One of the first applications was the scanning of humans for the animation industry. The Head Scanner was developed by Cyberware laboratories of Los Angeles.

Figure 2 Head scanner [22]

By the mid – nineties the full body scanner was developed. This is considered the point where 3D Scanners appeared.

Figure 3 Full body scanne r[22]

In 1994 serious progress was made with the lunch of REPLICA which allowed fast, highly accurate scanning of very detailed objects.

Figure 4 REPLICA [22]

Two years later in 1996, 3D scanners combined the technologies of a manually operated arm and a stripe 3D scanner and created the ModelMaker (shown in the figure below). This was a fast and flexible system which is considered to be the first Reality Capture System. It can produce complex models and texture them with color. With the lunch of REPLICA 3D models could be produce in minutes.

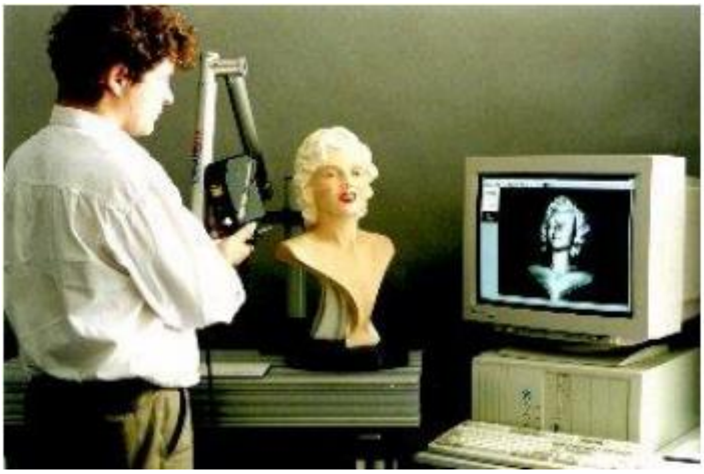


Figure 5 ModelMaker [22]

At this point laser scanners can be classified in 3 large categories:

1. Airborne Laser Scanners (ALS)
2. Terrestrial Laser Scanners (TLS)
3. Hand-Held Laser Scanners

Further we will briefly explore the first two types of laser scanners with the focus being on the third (Hand Held laser scanners) since it was the type used for the collection of this project's data.

1.4.1 Airborne Laser Scanners (ALS)

In 1994, a new airborne terrain modelling technology became available to the surveying industry.

This laser technology evolved as an alternative to traditional photogrammetric acquisition.

The ALS consists of three fundamental components:

1. The position of the aircraft is determined by kinematic dual frequency GPS, usually at 1 second epochs.
2. The orientation of the aircraft is monitored by a sensitive Inertial Reference System at a rate of 50 times per second.
3. The terrain measurement device emits discrete laser beams (5000 to 25000 per second), measuring the time needed for the beams to reflect form the ground back to the aircraft.

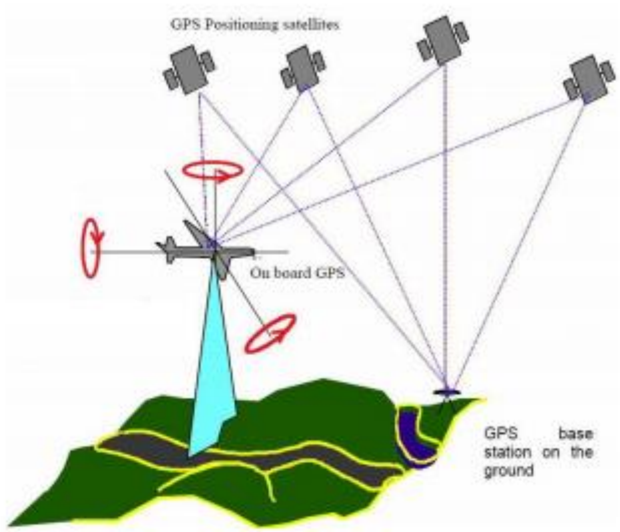


Figure 6 ALS system fundamental components [22]

A rotating mirror directs the laser beam in a swathe across the ground, then the post processing software combines the scanner's position, its orientation and the distances measured by the laser beams to compile a digital elevation model.

The operational parameters of laser frequency, swathe width, flying height and aircraft velocity can be tailored to meet the optimum point density for each project. The point spacing can vary depending on the project. It can range from under 1 meter up to 10 meters covering a swathe width of up to 700 meters. The laser emitted by the scanner is not in the visible spectrum and is eye-safe. It has a wavelength of 1.04 microns.

1.4.2 Terrestrial Laser Scanners (TLS)

Terrestrial Laser Scanning uses the same principle as Airborne laser Scanning, except that it is grounded. With the scanner being on the ground it gives the user certain advantages for surveying objects from multiple angles.

A TLS system can measure several thousand points per second allowing data sets to be collected in excess of what would be obtained by traditional surveying or photogrammetric techniques. It is most useful for surveying small (relative to ALS) irregular objects such as buildings, natural land formations or mining operations.

Terrestrial Laser Scanning has already found its place between the standard technologies for object acquisition. The laser scanner can be described as a motorized total station which measures automatically all the points in its horizontal and vertical field. The device records the distance from the point to the laser scanner along with the horizontal and vertical angle for each surveyed point. Therefore, the spatial coordinates can easily be computed for each point relative to the scanner's position.

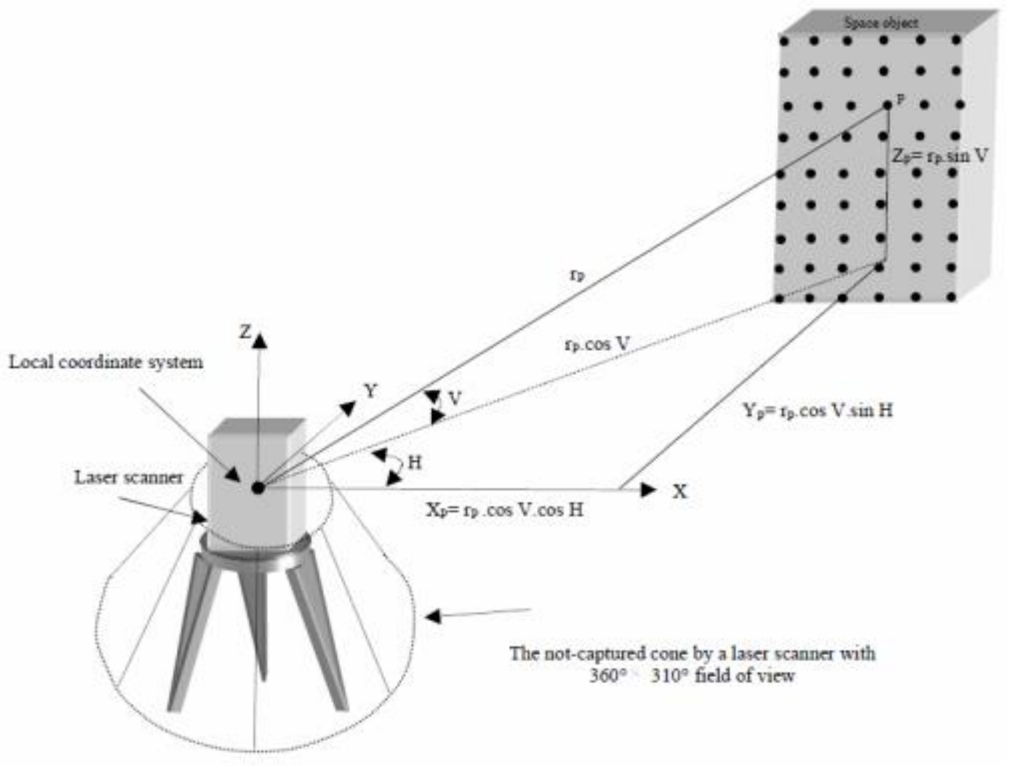


Figure 7 Surveying points with TLS [22]

This means that at one position of the TLS, a dense point cloud is immediately delivered.

Along with the points coordinates, the laser scanner also measures the intensity value of each point. Intensity is defined as the measure of electronic signal strength obtained by converting and amplifying the backscattered optical power. These measurements are commonly used to support the visual analysis of the point cloud. It can also be used in more sophisticated applications such as registration and classification by surface material.

1.4.3 Hand-Held laser Scanners

Hand – held laser scanners create 3D images through the triangulation mechanism: a laser dot or line is projected onto an object from a hand-held device and a sensor measures the distance between the device and the surface. Data is collected in relation to an internal coordinate system, therefore, to collect data when the scanner is in motion the position of the scanner must be determined. This is done by the scanner using reference features on the scanned surface or by using an external tracking method.

External tracking usually means some form of laser tracker (which provides the sensor's position) along with an integrated camera (which is used to determine the orientation of the scanner). Another option is the use of 3 or more cameras which provide a complete 6 degrees of freedom to the scanner. Both techniques tend to use infrared light emitting diodes attached to the scanner, which are seen by the camera(s) through filters providing resilience to ambient lighting.

Data is collected by an internal computer and recorded as points within a 3D space. By post processing this data can be converted into a triangulated mesh and then a CAD model.

GeoSLAM ZEB 1

The device used for data gathering in this project is a GeoSLAM ZEB I.

Figure 8 GeoSLAM ZEB-REVO

Specifications

Data Acquisition Speed: 43,200 points/sec
3D Measurement Accuracy: +/- 0.1% (typically)
Maximum Range: Up to 30m (15m outdoors)
Laser Safety Class: Class 1 Eye Safe
Angular Field of View: +270 x ~100 degrees
Weight of Scanner Head: 665g
Dimensions of Scanner Head: 60 x 60 x 360 mm

The device was lunched at the beginning of 2013. It is mostly used for surveying inside buildings, documenting car crashes or crime scenes, mapping the inside of mines and for facilitating emergency planning.

It can be adjusted to different scenarios, since the device can be hand held or mounted on a vehicle.

As far as user experience is concerned, the Geo SLAM is extremely user friendly. For example, we only had half an hour of training with one of COWI's personnel and another hour of personal research on the internet before the scan.

The scanning process itself is simple and quick. It takes about 15 minutes to set up and initialize the device before you start the scan.

For this project, the scanned building has 3 levels, which we scanned one at a time. Because the device has no built-in GPS system, the scanned points are positioned based on the user's movement and the previously scanned points, therefore, the scan needs to end in the same place it started. Also, to increase the quality of the data it is encouraged to loop as much as possible inside the original loop. To do this efficiently the user needs to plan the route beforehand and to open all the doors before the scan starts.

During the scan the user needs to keep an eye on the blinking light on the device. Since it has no screen, the light is the only way for the device to send information to the user during the scan. It does so by changing color and/or blinking intervals. During the scan the light can be either green, yellow or red. If the light is red it means something is wrong, if it's yellow it's working but not in optimal conditions and if it is green it means that everything is working fine.

Figure 9 GeoSLAM ZEB 1 work flow [3]

The final step is to download the data from the GeoSLAM. This is done at the end of the scan when the user completed the loop and it is in the same spot where the scan started. The download process is straight forward. The user only needs to connect an USB stick to the device and to wait for the blinking light to signal that the download is complete (all the code messages can be found in the user manual).

After the download is complete, the user need to turn the device on another side to start a new scan.

The biggest down side of using the GeoSLAM is that most of the process is a black box which is not great for academic purposes since we can not explain the whole process in detail. Processing data from the GeoSLAM is no difference. After downloading it from the device, we uploaded it on the company's platform where they process the data. When the processing is complete the user needs to pay an amount of credits (depending on the surface of the scan) to be able to download the data from the cloud.

As user you have little to no control over your data. The processing is done by an algorithm which we have no access to, so it is impossible for us to know how it works. If for some reason the data downloaded after the processing is not good (assuming there were no errors in the scanning process) you can flag it and they will do the processing all over again. If this is the case, the processing is no longer done by the algorithm alone, it is supervised by a human user.

GeoSLAM ZEB – REVO RT

The latest hand-held scanner from GeoSLAM is GeoSLAM ZEB – REVO RT(real time) which was launched in 2017.

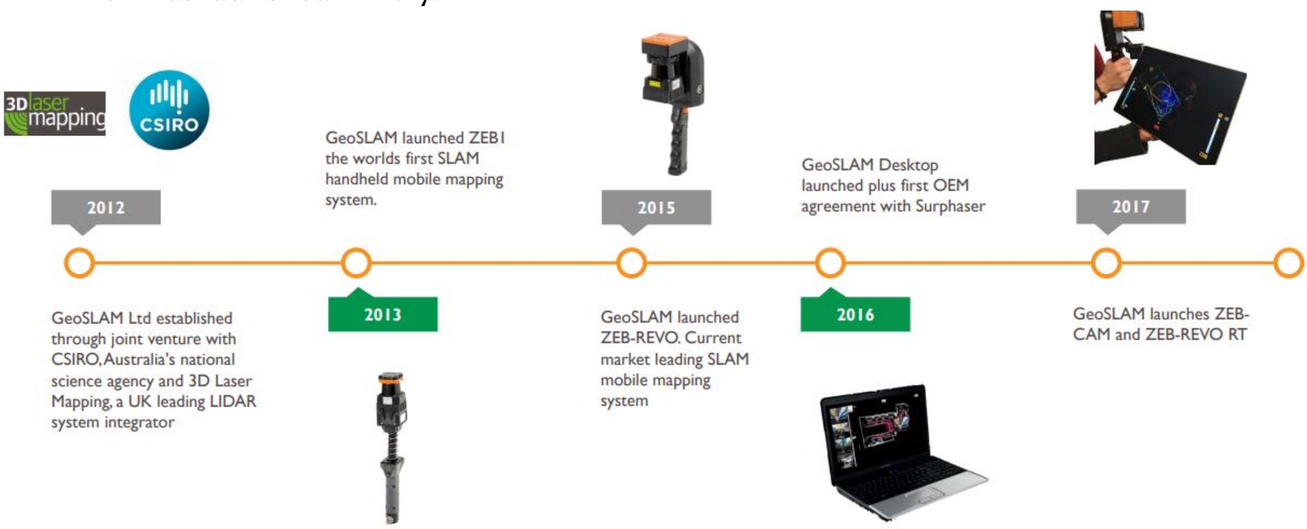


Figure 10 GeoSLAM product launch timeline [3]

A very important update to the older version is the Real Time function, which allows the point cloud to be built while the scanning process in ongoing. This eliminates the post processing step leading to a more efficient work flow.

Figure 11 GeoSLAM ZEB – REVO RT work flow [3]

Another important factor is that with the new model the user can view the data as it is being scanned in the field.

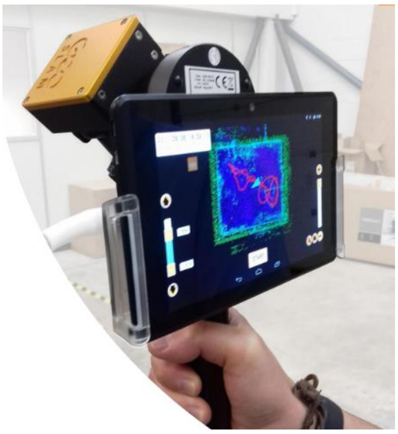


Figure 13 GeoSLAM ZEB – REVO RT [3]

Having a screen is a big improvement compared to the device used for the data collection of this project which could only transmit data to the user using a blinking light.

Last but not least, the latest model doesn't need the user to create the motion required for the scan manually. Now the device's laser sensor rotates itself while mounted on an inertial measurement unit.

Figure 14 GeoSLAM ZEB – REVO RT [3]

Due to this improvement the angular field of view of the scanner also got better. Due to its circular movement the field of view changed from 270 x 100 degrees to 270 x 360 degrees.

Conclusion

The reason laser scanning was not used earlier is because the data generated by laser scanning consists in big files containing point clouds and other information which were just too big for computers to handle until more recently. Basically, laser scanning is one of those technologies which could not advance until band width, hard drive storage and computational power became more affordable. Even today point clouds usually have to be delivered on external hard drives because the files are too large to e-mail or FTP (File transfer protocol). Laser scanning same as the GPS was only used by the military, surveyors, pilots and other specialized personnel in its early days but, lately has become a more common thing. For example: the "Xbox Kinect" is a form of laser scanning that scans the players in real time, recording their shape and movements and from this, creates a virtual reality.

Laser scanning is a very straight forward science. It uses 3D points to define real world surfaces that are being scanned. The scanned object can be anything from a car to an industrial plant or even a person. However, the art of 3D scanning is in the final deliverable. The point cloud can look beautiful in its native software, but upon delivery the client may find it unusable, because of its size. A typical point cloud can be anywhere from 10 to 100 Gbs, making most CAD software inoperable. The real power of scanning is when you can utilize the point cloud and the as-build model derived from the point cloud in your current software system.

1.5 How can a BIM be created from a point cloud?

When creating a BIM from point clouds, there are two possible paths.

- Direct modelling
- Parametric modelling
    - Parametric – generative modelling
    - Parametric object - oriented modelling

1.5.1 Direct modelling

The direct modelling approach starts with a point cloud imported into the software which uses the input as a base to vectorize the required profiles for the 3D model.

The modelling time depends on the number of sections that need to be vectored and the complexity of the object. It is mandatory to extract as many "slices" as the number of different profiles.

Creating a complex 3D model with this method, requires a lot of time due to the higher number of horizontal and vertical sections.

The 3D surfaces also require several passages and commands from the user and the results can be tricky.

At the end of the workflow the created element is unique and can't be easily adapted to similar parts of the building (if there are any). The model is then fitted of the survey data and because of this, similar components require separate modelling from scratch.

Because of all these reasons, the direct modelling approach creates a bottleneck effect therefore it is advised to use the next method instead.

1.5.2 Parametric modelling

An alternative to the Direct Modelling approach is the Parametric Modelling. This method allows a workflow through pre-established models, that can be edited in any step of the process by inserting new parameters. This technique is mostly used when constructing new building, however it has great potential also when it comes to existing ones.

Parametric modelling can be classified in two categories:

1. **Parametric generative modelling.**

The generative modelling approach is based on the use of algorithms therefore, it is possible to refer to it as algorithmic modelling.

The goal is to produce a general procedure as solution to similar problems. This is the most relevant aspect of this method because it would create an abacus of algorithmic and parametric models that describe most of the similar elements of the building. It could be a database, for example, composed by virtual elements that correspond to the shapes of the construction and every element of the model corresponds to all elements with similar features but different dimensions.

When building big scale models, it is necessary to take into consideration the diversity of the elements and follow their shape. To do so, a theoretic model is created, built on the primitive shapes and applicable to all other similar elements.

The parametric generative modelling method consists in 3 phases:

1.1. Deconstruction phase.

During this first phase the object is studied to identify its generative shapes and the ways in which it can be geometrically built.

The next step is the extraction of the profiles that will be later used to generate the model's surface from the point cloud. This is done exactly the same as in Direct modelling, the main difference being that in this case the user only needs to do this step once. These profiles will later become the input data of the generative modelling.

1.2. Reconstruction phase.

In this phase the parametric model is created and the command sequence is designed.

This means that the connections that elaborate the input data are created and the final model results as output data.

1.3. Concretization phase.

The final step is a transition between the parametric model and the modelling environment.

By turning the model into a mesh, the user can overlap it over the point cloud to verify the model's fitting over the survey data. It is necessary that the difference between the model and the point cloud is a value within a tolerance threshold.

By using this method, the user doesn't need to model each element individually from scratch. Once the first model is built he/she can immediately extract the others, simply by assigning the correct input data.

2. **Parametric object – oriented modelling.**

This method refers essentially to the BIM working process, as it uses ***standard*** constructive elements to build the model. The word standard is used to highlight that this method is mostly used when constructing new buildings. When it comes to existing buildings we have to take into consideration the unicity of the building's elements. Because of this, specialized software companies started to develop functionalities that can handle problems coming from existing structures such as managing the point cloud, assigning specific information to key points and modelling of complex shapes.

When dealing with existing buildings, the starting point is not a project, but a geometrical survey and the BIM is inserted in the lifecycle of the building.

The general target is to investigate the possibility of using the parametric modeling, that is usually applied to new constructions to an existing one.

Chapter 2 - Data acquisition

2.1 Scanned data

The following chapter describes the process of collecting data, information about the hardware used for the scan and some information about the post processing of the collected data.

The first problem encountered was to choose a building to model. This meant that sever factors were taken into consideration:

- Access to the building
- The size of the building
- The shape of the building

Having discussed with our supervisor we decided that the perfect building for the job is one of Aalborg University's buildings.

Figure 15 Side view of the scanned building

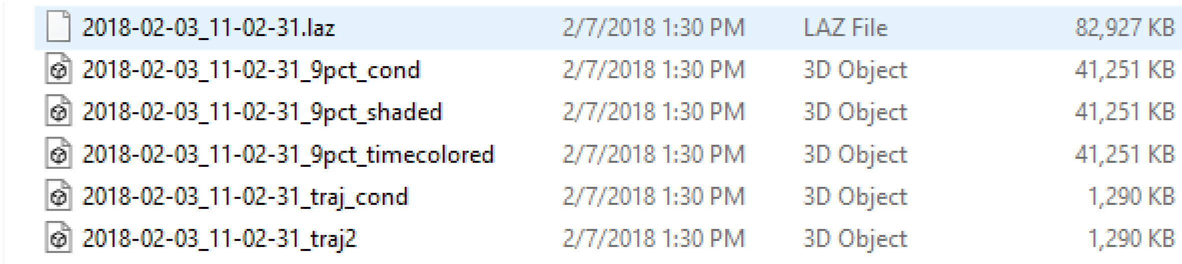
This building perfectly matched to all our needs, since getting access to the building was easy, the shape of the building is rectangular and it is large enough for us to gather a significant amount of data.

The next step was to decide how we are going to approach the scanning process. A regular laser scanner means that we would have to scan each room of the building individually, which is tedious work, so we discarded this option quickly and started looking for alternatives. This led us to the GeoSLAM web page [3]. Using GeoSlam seemed to fit perfectly with the task we had in front of us. The issue was that the university did not own a GeoSlam device and we originally had no idea how to use it. With the help of our supervisor we contacted COWI one of Denmark's largest surveying companies and asked for their help. Fortunately for us, the people from COWI were eager to help, so they provided us with the equipment and, also explained how the device works.

As mentioned before, the data for this project consists of four individual point clouds. Three of these contain the interior of the building while the fourth is a representation of the exterior of the building.

Forward the data will be shortly described as a whole. Then, every point cloud will be explored individually.

After the processing is done and the payment is issued, the user can download the data from the cloud. It comes as an archive containing several elements:



2018-02-03_11-02-31.laz	2/7/2018 1:30 PM	LAZ File	82,927 KB
2018-02-03_11-02-31_9pct_cond	2/7/2018 1:30 PM	3D Object	41,251 KB
2018-02-03_11-02-31_9pct_shaded	2/7/2018 1:30 PM	3D Object	41,251 KB
2018-02-03_11-02-31_9pct_timecolored	2/7/2018 1:30 PM	3D Object	41,251 KB
2018-02-03_11-02-31_traj_cond	2/7/2018 1:30 PM	3D Object	1,290 KB
2018-02-03_11-02-31_traj2	2/7/2018 1:30 PM	3D Object	1,290 KB

Figure 16 Elements form ground floor archive

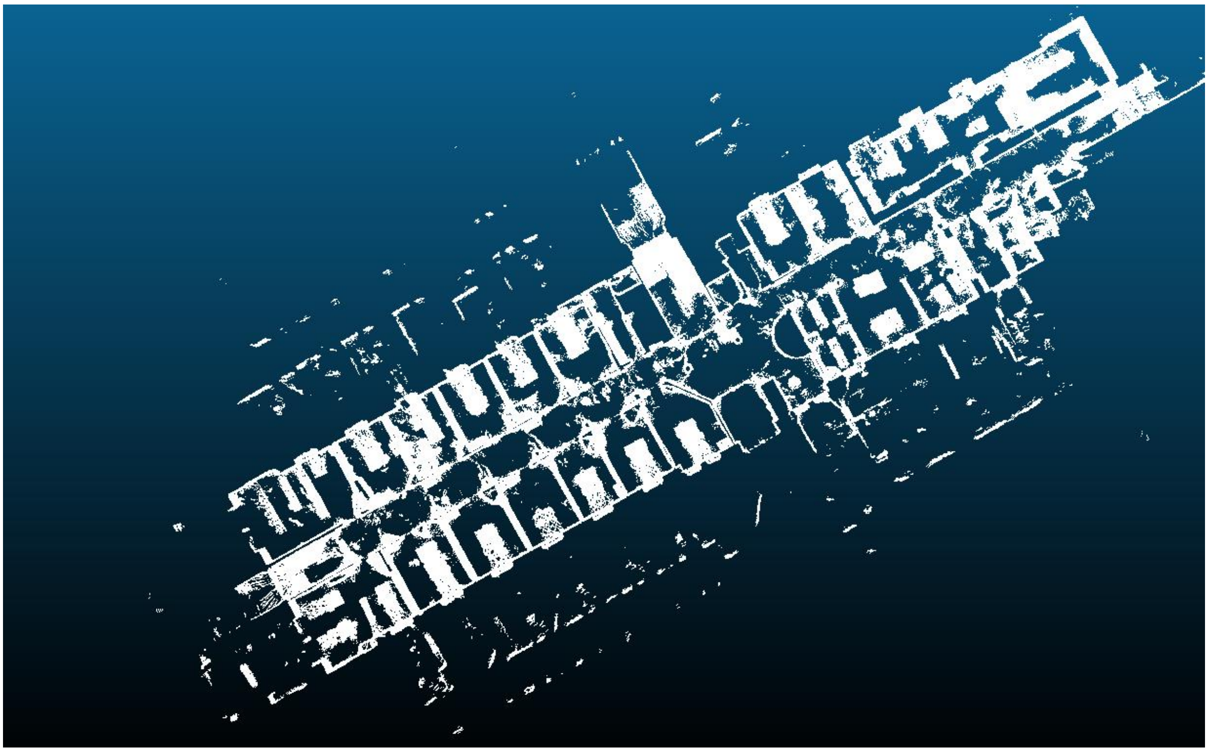


Figure 14 Ground floor point cloud raw

The first element is a .LAZ file containing the entire point cloud.

The point cloud contains around 29 million points. As it can be seen in Figure 16, there are some points that are not part of the building. These points are the result of the building having a large number of windows. They will be later deleted because they are not representative for this project.

The next three elements represent a lesser version of the point cloud, containing about 9% of the original cloud. Each element represents different aspects of the point cloud such as the time stamp for the collected points, the SLAM condition colored or the ambient occlusion. These output elements were selected by the user from a larger list of possible outputs.

Option	**Descriptor**	**Description**
% of points	_%*X*	Where X is the percentage of points selected Omitted if set to 100%
Special decimation	_*XX*mm	Where XX is the spatial decimation selected Omitted if no spatial decimation selected
Point colour	_height _time _shade _cond _shape	Height coloured Time coloured Shaded coloured (ambient occlusion) SLAM condition coloured Shapecoloured Omitted if set to None
Timestamp	_ts _tss	World (UNIX) time timestamp Scan time timestamp Omitted if set to None
Normals	_norm	Normals included Omitted if set to None

Figure 15 List of possible output files from the SLAM software [4]

The _cond file from Figure 18 shows the condition of all the surveyed points using a blue-red scale, with blue meaning that the points are in good condition while red means that the condition of the points is poor.

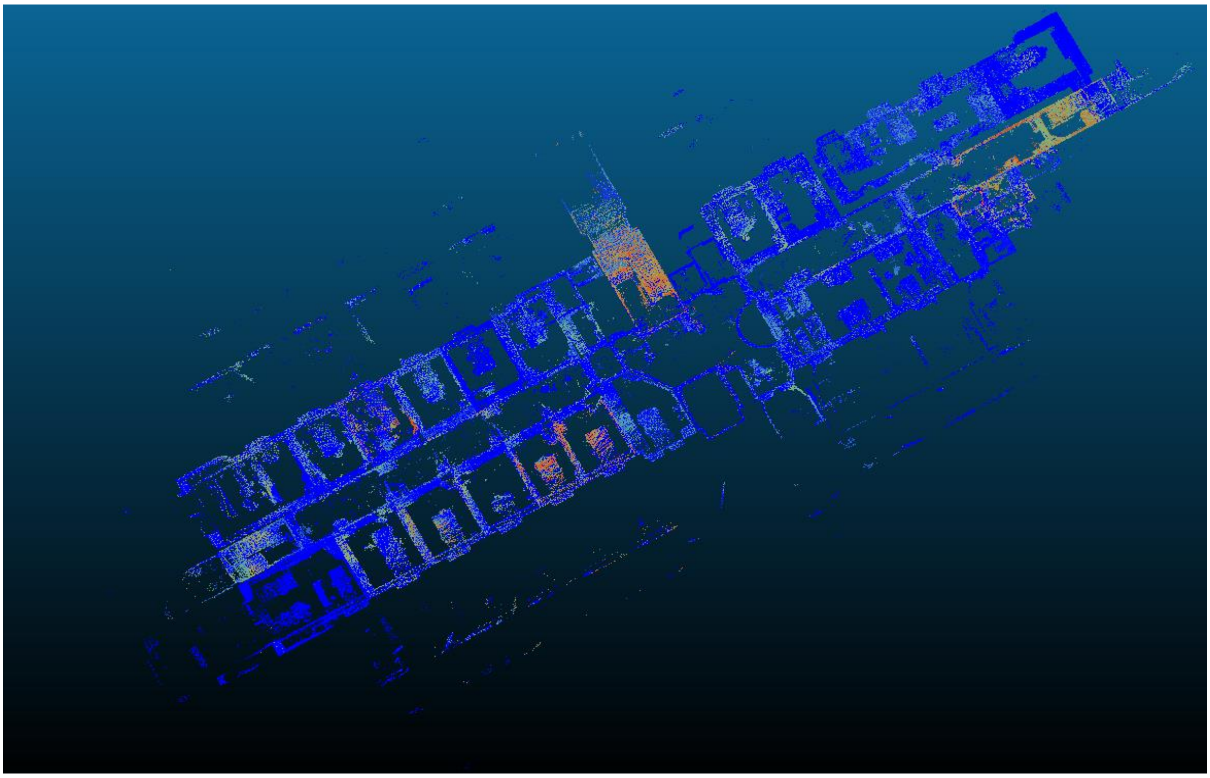


Figure 16 Visual representation cloud's condition of the ground floor of the point

The _shade file From Figure 19 shows the ambient occlusion of the points with a greyscale.

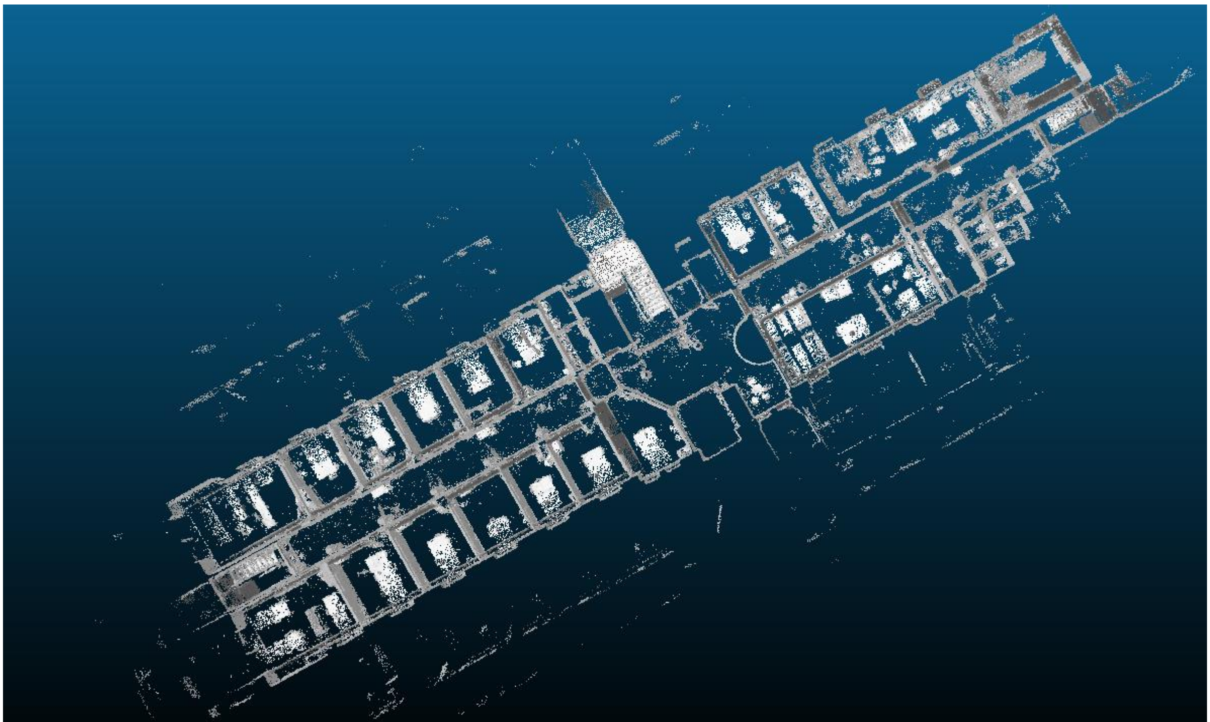


Figure 17 Visual representation of the ambient occlusion on the ground floor

The _timecolored file from Figure 20 shows the time stamp of the collected points by a colored scale going from red on the first scanned points to blue on the least.

Figure 18 Visual representation of the timecolored file of the ground floor

The last two files from Figure 17 represent the trajectory followed by the surveyor when the points were collected.

Column	Descriptor	Description
1	time	Time in UNIX time
2	x	Scanner x coordinate in metre relative to scan origin
3	y	Scanner y coordinate in metre relative to scan origin
4	z	Scanner z coordinate in metre relative to scan origin
5	q1	Orientation quaternion
6	q2	Orientation quaternion
7	q3	Orientation quaternion
8	q4	Orientation quaternion
9	Userfields	Currently unused

Figure 19 Information that can be extracted from the trajectory file [4]

Figure 21 shows the types of data that can be extracted from the trajectory file. It contains information about the user's position and orientation at any given time during the scan.

The traj_cond file from Figure 22 displays the condition of the scan during the survey at any given time with a color scale ranging from blue to red, with blue showing that the condition of the scan was good and red meaning that it was poor.

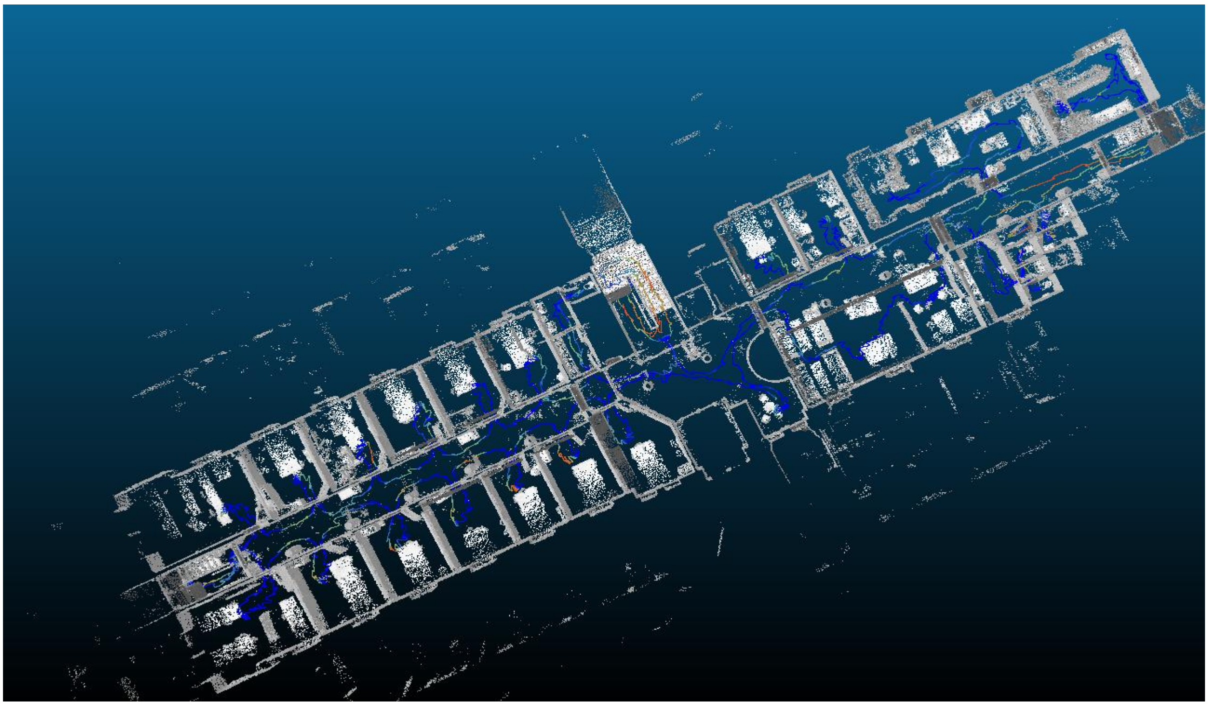


Figure 20 Visual representation of the traj_cond file

The last output file is _traj2 from Figure 16. It contains the time colored version of the trajectory ranging from blue at the start of the scan to red at the end.

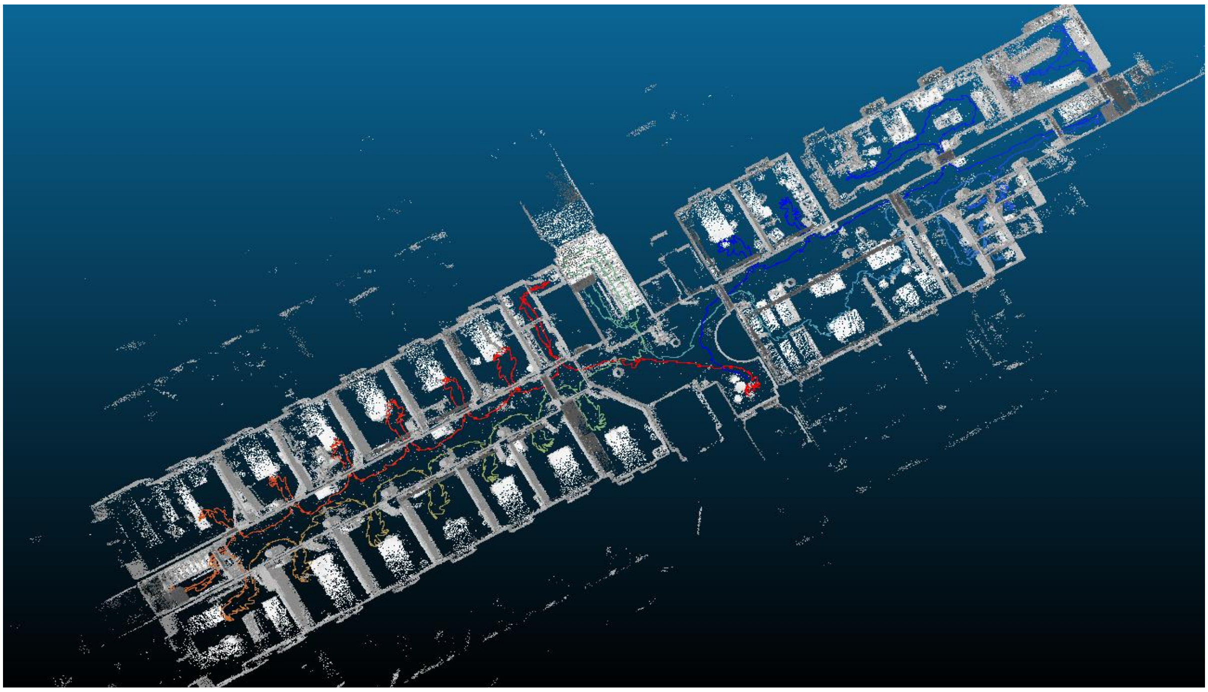


Figure 21 Visual representation of the time colored trajectory

These were the files output from the ground floor scan. The same elements can be found in the other three scans but, to avoid repetition, they will not be discussed individually.

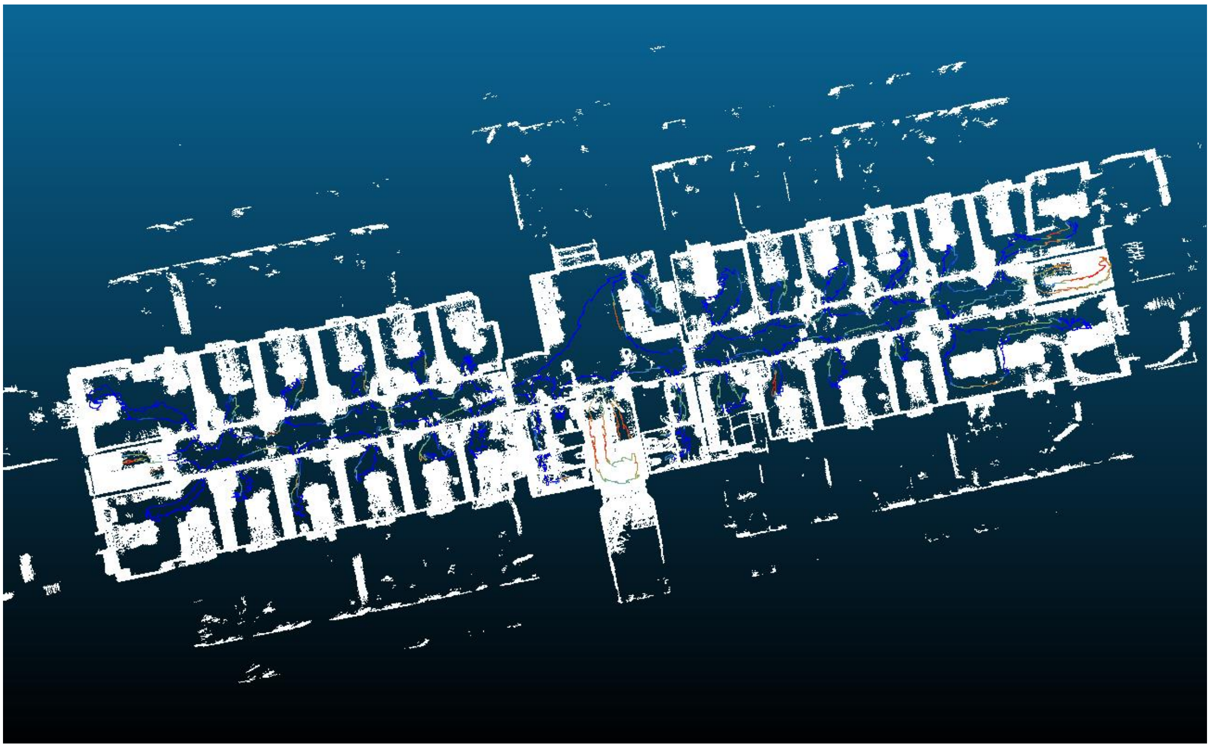


Figure 22 Visual representation of the first floor and the trajectory

Figure 24 shows the top view of the scan from the building's first floor and the trajectory of the survey. The point cloud is a little bit bigger than the one from the ground floor. It has roughly 32 million points (3 million more than the ground floor). This is mostly because there are more rooms at the first floor than there were at the ground floor.

It can be observed in Figure 24 or in any other visual representation of the ground floor that the building contains three stair cases. These will be used later on in the project to align the scans one on top of another for the final building model. Because of this, the stair cases are the common points between the ground floor and the first floor (Figure 25). The same situation will be found between first floor and second floor.

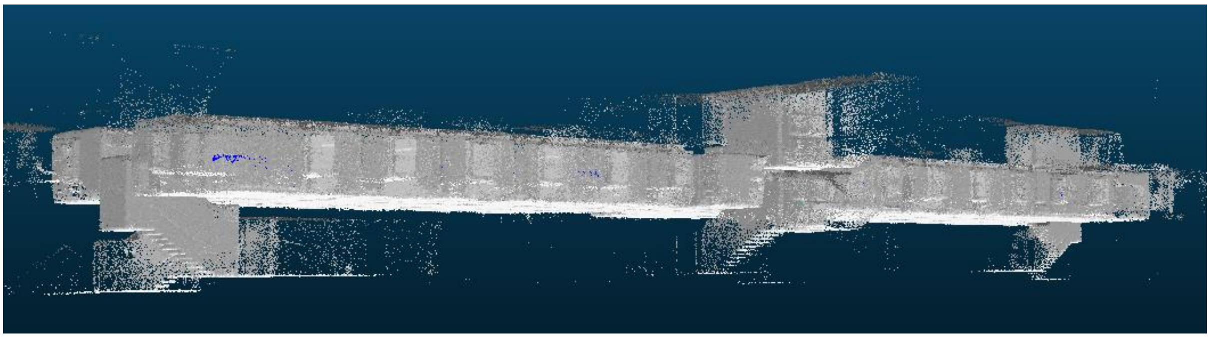


Figure 23 Side view of the first floor

The point cloud from the second floor of the building is the smallest interior point cloud because there are few rooms and a lot of open space as it can be observed in Figure 26.

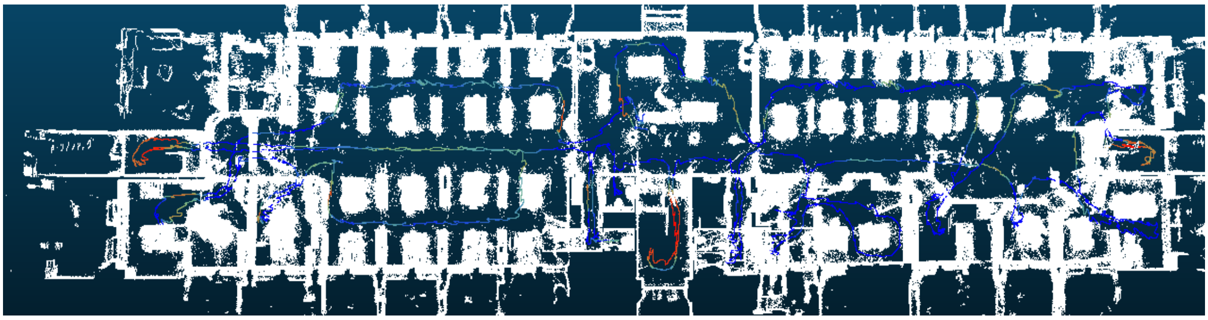


Figure 24 Visual representation of the second floor scan and the trajectory

Also Figure 27 shows the common points between the first floor and second floor.

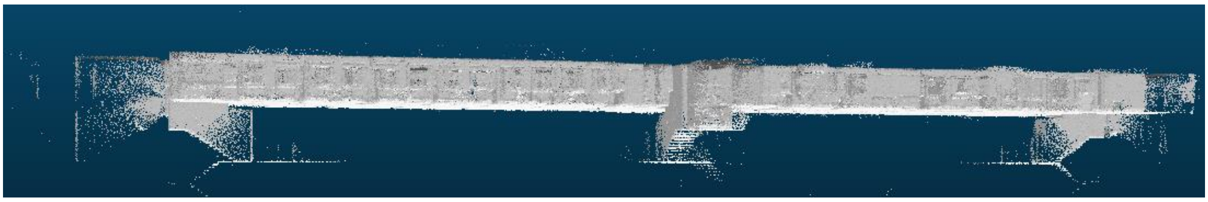


Figure 25 Side view of the second floor

For a complete building model, the exterior of the building was also scanned. The point cloud containing the exterior is the smallest one having around 8 million points. Figure 28 shows a top view of the scan along with the trajectory.

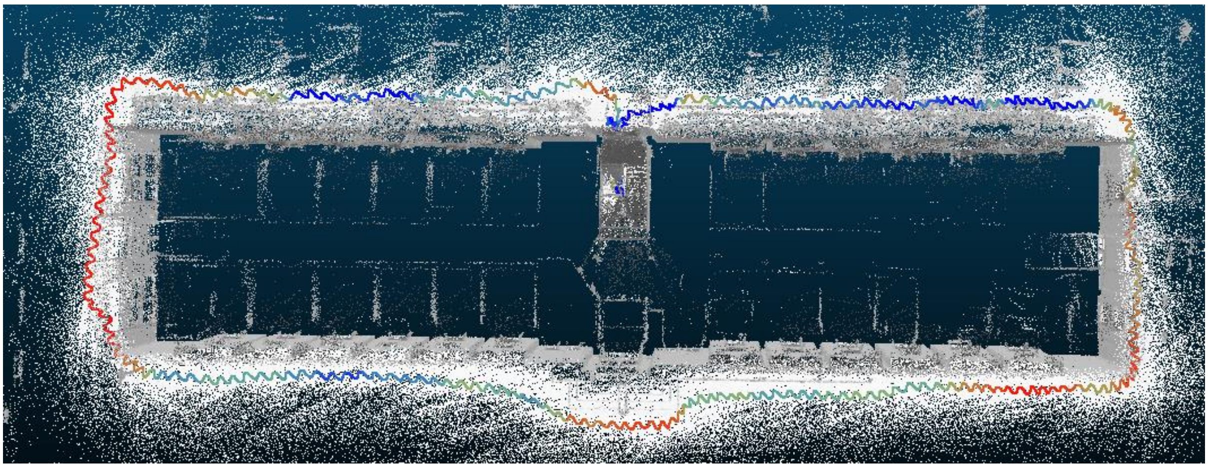


Figure 28 Top view of the exterior and the trajectory

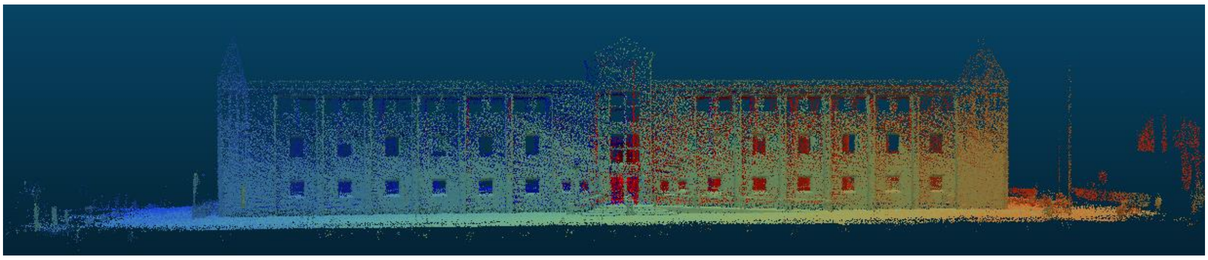


Figure 29 Side view of the exterior

2.2 Total station measurements

Further in this chapter, the process of collecting data with the Total station will be described. These measurements hold two purposes:

1. Aligning the point clouds.
   As described in chapter 1, the GeoSLAM used for scanning the building doesn't have any GPS system incorporated, therefore all point clouds created have no relation to each other and have coordinates only in a local coordinate system. Another problem caused by the scanner having no GPS system is that all point clouds have a different orientation so they don't overlap without post processing. The most common solution is to conduct a survey with a Total Station, survey the points of interest (points of interest are common points of two or more point clouds later used for alignment).
2. Accuracy check at the end of this chapter.
   The total station measurements will also be used in the last chapter of this paper to check the accuracy of the BIM. This process will be described in a further chapter. To conduct the survey, we used a network around the university created by our professors.

Figure 30 Overview of the support network around Aalborg University

The network consists of 16 points, but for the purpose of this project we are only going to use 4 of them (6003, 6010, 6009, 6004) because they are the closest to our building and their geometry includes our building of interest.

Point number	E	N	H
6000	395516.8	1325346	1.707
6001	395578.7	1325318	1.777
6002	395643.2	1325289	1.806
6003	395698.2	1325267	1.937
6004	395827.2	1325220	1.739
6005	395875.3	1325212	1.713
6006	395853.8	1325074	2.547
6007	395851.7	1325035	1.867
6008	395750.3	1325080	1.482
6009	395764.5	1325108	2.434
6010	395661.8	1325174	1.549
6011	395638.6	1325112	1.446
6013	395524.3	1325179	1.604
6014	395485.6	1325274	1.469
6015	395526.5	1325224	1.636
6016	395604.8	1325207	1.645

Figure 31 table of coordinates for the support network

For this survey the University provided us with a Leica TS09 Total station which we used to collect our data. The total station has an angular measurement accuracy of the horizontal and vertical angle of 1" (0.3 mgon) / 2" (0.6 mgon) 3" (1 mgon) / 5" (1.5 mgon). The angular measurements are done using one of the following methods: absolute, continuous, or diametrical and angular compensation is done using the quadruple axis compensation.

Figure 32 total station used for the survey

The distance measurements surveyed with the reflector less function can be done on any surface and the accuracy is 2mm + 2ppm, while the measurement time is 2.4 seconds. The laser dot size at 30 m is approx. 7 x 10 mm and approx. 8 x 20 mm at 50 m.

For the measurements, we used two different types of prisms and the reflector less technique.

For the exterior a 34.4 mm constant Leica GPH1P round prism was used for the ground floor and the reflector less technique for upper floors.

Inside, a Leica GMP111 was used. This is a smaller prism with a constant of 16.9mm.

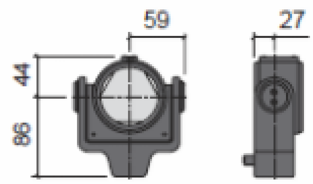
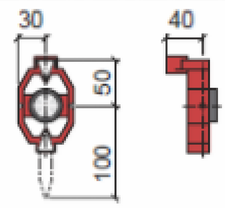
Prism Model	Actual Offset	
GPH1P	-34.4	
GMP111	-16.9	

Figure 33 Prism types and constants

The main reason for the prism change was that the GMP111 is smaller and much more comfortable to work with inside compared to its larger counterpart.

For this survey two jobs were created. One for the exterior of the building and the other for the interior.

The data is processed using TMK, a software created by our professors at Aalborg University, which is used in MatLAB. The process is a fairly simple one, first the raw data from the total station in converted from a .GSI file to an .OBS file.

```
*410003+0000000000000003 42....+0000000000001001 43....+0000000000000014 44....+0000000000001650
*110004+0000000000000000 21.322+0000000039999820 22.322+0000000009997213 31..00+0000000000099809 51..1.+000000000001+018 87..10+0000000000002000 71....+0000000000000014
*110005+0000000000000001 21.322+0000000039212533 22.322+0000000009991175 31..00+0000000000102452 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110006+0000000000000002 21.322+0000000039148831 22.322+0000000009986294 31..00+0000000000098353 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110007+0000000000000003 21.322+0000000039042963 22.322+0000000009986522 31..00+0000000000091857 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110008+0000000000000004 21.322+0000000039037814 22.322+0000000009986292 31..00+0000000000091386 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110009+0000000000000005 21.322+0000000038907132 22.322+0000000009983994 31..00+0000000000084873 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110010+0000000000000006 21.322+0000000038831750 22.322+0000000009983909 31..00+0000000000081429 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110011+0000000000000007 21.322+0000000038819555 22.322+0000000009983922 31..00+0000000000080969 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110012+0000000000000008 21.322+0000000038784281 22.322+0000000009983860 31..00+0000000000081057 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110013+0000000000000009 21.322+0000000038748560 22.322+0000000009981234 31..00+0000000000077993 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110014+0000000000000010 21.322+0000000038697418 22.322+0000000009981312 31..00+0000000000074430 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110015+0000000000000011 21.322+0000000038587904 22.322+0000000009981301 31..00+0000000000070489 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110016+0000000000000012 21.322+0000000038548923 22.322+0000000009975213 31..00+0000000000070621 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110017+0000000000000013 21.322+0000000038436720 22.322+0000000009981224 31..00+0000000000067268 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110018+0000000000000014 21.322+0000000038396875 22.322+0000000009981286 31..00+0000000000067394 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110019+0000000000000015 21.322+0000000038339243 22.322+0000000009981314 31..00+0000000000064338 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
*110020+0000000000000016 21.322+0000000038316197 22.322+0000000009977902 31..00+0000000000063906 51..1.+000000000001+000 87..10+0000000000002000 71....+0000000000000000
```

Figure 34 Contents of GSI file

```
TC110
TMK, Konvertering fra GSI-format til TC110-format           2018- 5- 1 10:56:41
GSI-fil: F:\Docs\TMk\EXT.GSI

L=   3: OPSTL =         6003         KODE =   14                 ih =      1.650M
L=   4: PKT =           6010 Hz =  399.9982G   V =   99.9721G  Sd =    99.809M  K =   14  sh = 2.000M
L=   5: PKT =              1  Hz = 392.1253G   V =   99.9117G  Sd =   102.452M  K =    0  sh = 2.000M
L=   6: PKT =              2  Hz = 391.4883G   V =   99.8629G  Sd =    98.353M  K =    0  sh = 2.000M
L=   7: PKT =              3  Hz = 390.4296G   V =   99.8652G  Sd =    91.857M  K =    0  sh = 2.000M
L=   8: PKT =              4  Hz = 390.3781G   V =   99.8629G  Sd =    91.386M  K =    0  sh = 2.000M
L=   9: PKT =              5  Hz = 389.0713G   V =   99.8399G  Sd =    84.873M  K =    0  sh = 2.000M
L=  10: PKT =              6  Hz = 388.3175G   V =   99.8391G  Sd =    81.429M  K =    0  sh = 2.000M
L=  11: PKT =              7  Hz = 388.1956G   V =   99.8392G  Sd =    80.969M  K =    0  sh = 2.000M
L=  12: PKT =              8  Hz = 387.8428G   V =   99.8386G  Sd =    81.057M  K =    0  sh = 2.000M
L=  13: PKT =              9  Hz = 387.4856G   V =   99.8123G  Sd =    77.993M  K =    0  sh = 2.000M
L=  14: PKT =             10  Hz = 386.9742G   V =   99.8131G  Sd =    74.430M  K =    0  sh = 2.000M
L=  15: PKT =             11  Hz = 385.8790G   V =   99.8130G  Sd =    70.489M  K =    0  sh = 2.000M
L=  16: PKT =             12  Hz = 385.4892G   V =   99.7521G  Sd =    70.621M  K =    0  sh = 2.000M
L=  17: PKT =             13  Hz = 384.3672G   V =   99.8122G  Sd =    67.268M  K =    0  sh = 2.000M
L=  18: PKT =             14  Hz = 383.9688G   V =   99.8129G  Sd =    67.394M  K =    0  sh = 2.000M
L=  19: PKT =             15  Hz = 383.3924G   V =   99.8131G  Sd =    64.338M  K =    0  sh = 2.000M
L=  20: PKT =             16  Hz = 383.1620G   V =   99.7790G  Sd =    63.906M  K =    0  sh = 2.000M
L=  21: PKT =             17  Hz = 382.7864G   V =   99.7782G  Sd =    64.032M  K =    0  sh = 2.000M
L=  22: PKT =             18  Hz = 382.0129G   V =   99.7779G  Sd =    61.001M  K =    0  sh = 2.000M
L=  23: PKT =             19  Hz = 381.8042G   V =   99.7778G  Sd =    60.560M  K =    0  sh = 2.000M
L=  24: PKT =             20  Hz = 380.5165G   V =   99.7787G  Sd =    57.652M  K =    0  sh = 2.000M
L=  25: PKT =             21  Hz = 380.3281G   V =   99.7781G  Sd =    57.265M  K =    0  sh = 2.000M
L=  26: PKT =             22  Hz = 379.8160G   V =   99.7783G  Sd =    57.382M  K =    0  sh = 2.000M
```

Figure 35 Contents of the OBS file

After the conversion is complete, the OBS file must be custom edited, before the next step begins, by deleting unnecessary lines and changing the names of station points to match the point names from the support network.

Next, the edited .OBS file is loaded together with network's file of coordinates into TMK, where using the function *Detailpunktsberegning* (Detail Points Computation in eng.) the .OBS file is converted into an .KOO file containing the coordinates of all surveyed points in the network's coordinate system.

```
KOORDINATER
Punkt Objektkode Liniekode E-DKTM2-m N-DMTM2-m H-DVR90-m
         6003    14 9   395698.245 1325266.520       1.913
         6010    14 9   395661.772 1325173.646       1.573
            1    14 9   395672.833 1325167.273       1.706
            2    14 9   395674.804 1325171.005       1.775
            3    14 9   395677.839 1325176.962       1.758
            4    14 9   395678.015 1325177.404       1.760
            5    14 9   395681.160 1325183.387       1.777
            6    14 9   395682.799 1325186.572       1.769
            7    14 9   395683.038 1325186.995       1.768
            8    14 9   395683.463 1325186.825       1.769
            9    14 9   395684.452 1325189.759       1.793
           10    14 9   395685.671 1325193.163       1.782
           11    14 9   395687.534 1325196.852       1.770
           12    14 9   395687.942 1325196.658       1.838
           13    14 9   395689.605 1325199.812       1.762
           14    14 9   395690.007 1325199.634       1.761
           15    14 9   395690.959 1325202.598       1.752
```

Figure 36 Example of the .KOO file

The last step of this process is to convert the file of coordinates into an .DXF or .CSV file so that it can be loaded using CAD software. The conversion from .KOO to .DXF can also be done using TMK.

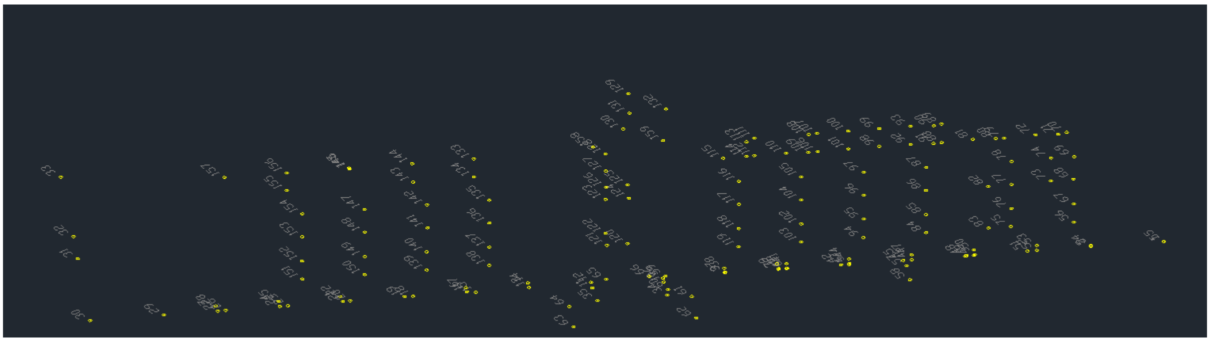


Figure 37 Side view of the survey in AutoCAD2019

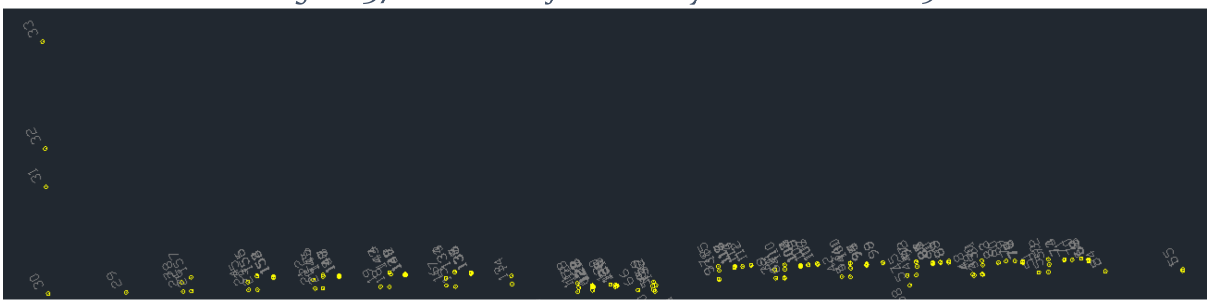


Figure 38 Top view of the survey in AutoCAD2019

Chapter 3 Problem statement

With an understanding of what BIM is and with the data collection in place, further exploration of means to create a digital model of a building is needed. For the creation of a model from a point cloud we need to get a grip on the current methods for geometry reconstruction. Then a model will be made and its accuracy assessed to measure the performance of the modelling approach.

The following questions are raised:

1. What are current methods to automatically create a solid object from point cloud data?
2. What are the steps in developing a model from cloud data?
3. How accurate is the model?

Chapter 4 - Automatic point cloud modelling

4.1 Automatic geometry reconstruction from point clouds

Although human vision and intuition can provide the most comprehensive understanding of the geometry and relationships between objects in complex scenes, especially at the levels currently required for BIM to be effective. Huge efforts are made to advance the field to an automatic process that can largely reduce the prerequisites for such a model. Both in industry and academia, different approaches are presented and discussed, from fully automated methods to semi-automatic ones, with varying degrees of success. Each aiming at obtaining a model as close as possible to the real asset.

The aim of this chapter is to explore some of the current approaches, both semi-automatic and fully automatic.

4.1.1 Commercial approaches

Currently the semi-automatic approaches revolve amongst well established industry leaders in CAD/CAE systems, in the form of plugins or different addons that aid in the construction of BIM. Image 1 below summaries some of their main features.

	Leica Cloudworx for Revit (Leica Geosystems, 2014)	**ImaginIT Scan-to-BIM**	**ClearEdge 3D Edgewise Building**	**Arithmetrica Pointfuse**
Automation (user input)	Semi-Automated pipe fitting only	Semi-Automated walls and pipes	Automated walls	Automated surfaces
Requires Revit?	Yes (but CAD versions exist)	Yes	Yes	No
Object-based parametric geometry produced?	Yes For pipes but really only acts as point cloud viewer for Cyclone data as a guide for modelling.	Yes As uses Revit family elements for fitted geometry.	Yes As elements detected in Edgewise are generated into Revit geometry in an RVT file export step.	No Produces 3D polygons/ surfaces that can be used as a guide for further modelling.

Figure 39 Comparison between commercial BIM solutions [10]

From the presented solutions, the two most promising are described below: ClearEdge3D and Scan-to-BIM.

ClearEdge3D classifies point clouds into surfaces that share coplanar points. With ClearEdge3D the user needs to pick ceiling and floor planes manually to constraint the software's wall detection. Then, the software automatically searches for the wall planes between the selected ceiling and floor which can then be brought up in Revit to construct the geometry. Figure 40 below shows the main stages of using this software.

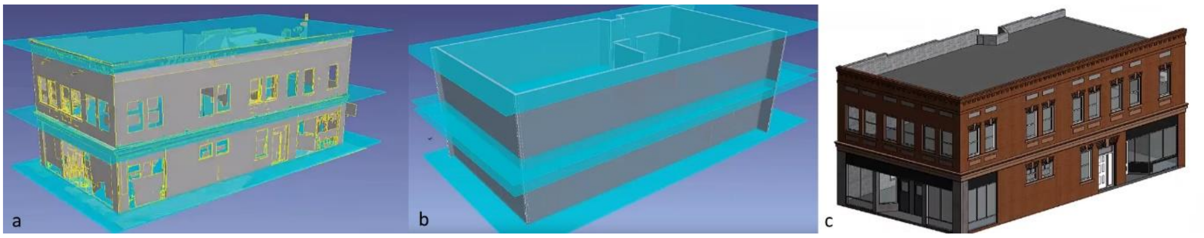


Figure 40 Clear Edge 3D a - selection of ceiling/floors; b - automatic wall detection; c - final model, after manual adjustments [8]

Scan-to-BIM is a Revit plugin developed by IMAGINiT Technologies that is perhaps the most successful solution in terms of deliverables. Simply put, Scan-to-BIM relies entirely on Revit for the handling of most tasks, such as loading the point cloud or using its geometry library. The main function is wall fitting, where the user selects 3 points to describe a plane from which a region growing algorithm detects the extents. Additionally, it provides some scan data handling tools as well as some fitting and detection algorithms.

As presented above, even the established companies with huge resource pools have a hard time developing a complete solution. At the time of this writing a complete commercial solution, even for the wall extraction, seems to not exist. The existing solution focuses on straight walls and pipes with the user having to define the planes of the ceiling and floor and manually verify each one of the extracted elements of the geometry. In addition, holes corresponding to windows and doors are hard to detect and further user input is needed to create an accurate model.

4.1.2 Academic approaches

The academic approaches consist mainly in the development of algorithms to speed up the modelling of geometry from point clouds. The research seems to have started around 1990 with the first attempts to automatically create geometry from photogrammetry data. Then it quickly moved towards airborne LIDAR data and later, façade reconstruction from terrestrial scanners and mobile mapping systems.

Research in the reconstruction of building interiors was born from the need for robots to understand the environment they are in, which is an important prerequisite for autonomous robots. This is an active area of research split into: segmentation and object detection. Both segmentation and detection blend together in multi stage methods for geometry reconstruction.

Segmentation of data is an established method from the field of computer vision, used to classify data with the same characteristics together. The main challenges of segmentation consist in developing an algorithm that can handle a broad range of geometries in a multitude of scenes. Here, the data quality has an important role by its nature. Current scanners can produce varying point densities, depending on the distance between the scanner and the object, which can affect the segmentation algorithms. Noise in the point cloud data is also an important factor, especially for the methods that take point normal into account when segmenting. It has been shown by George Vosselman (Vosselman et al., 2004) that noisy data affects the segmentation, especially for planar surfaces due to the normals varying too much.

Buildings are often composed of a multitude of primitive shapes (planes, spheres, cylinders, etc.). Amongst these, the most preponderant are planes, for which a multitude of segmentation methods where developed. Most notable are Principal Component Analysis (PCA), Least Squares (LS) and Random Sample Consensus (RANSAC). Both PCA and LS are reliable methods for planar segmentation, but they perform worse in certain situation than RANSAC. In contrast to the other two methods, RANSAC algorithms perform exceptionally well on datasets with high degrees of noise and the concept is simple to understand.

RANSAC, developed in 1981 by Fischler and Bolles, estimates the parameters of the searched model by random sampling a minimum number of data points which fit best with the model (eg. plane). Then it tests each random sample against each other to determine which one is closer to the sought model, the points that score the highest are labeled as inliers to the model and the rest are discarded (outliers). Over the course of the years there where different implementations of RANSAC, such as MLESAC (Torr and Zisserman, 2000) with a more robust scoring method and RMSAC (Rusu, 2008) that scores the inliers by squared distance.

Rusu et al. (2008) uses the modified version of RANSAC – RMSAC to detect and extract planes from the point cloud of a kitchen, for autonomous robot mapping. The detected planes are extracted and using a region growing algorithm it separates them into different entities. Then, a knob and handle identification and extraction phase for identifying if the detected objects are part of a certain object class (cabinet, door, fridge, etc.). The final step is to use an imposed or learned functional reasoning to assign candidates to different object classes. This work uses surface curvature and point normals to correctly identify horizontal planes (normals perpendicular to the Z axis) and vertical planes (normals parallel to the Z axis). To make the 3D volumetric geometry the boundaries of the detected objects are projected on the wall plane creating cuboids.

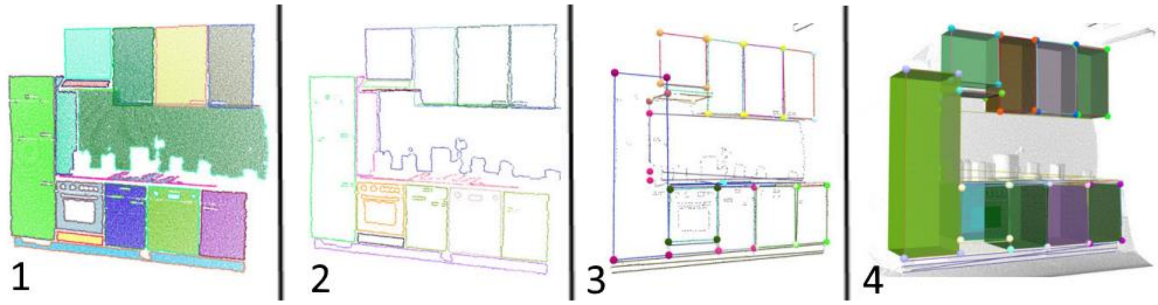


Figure 41. Boundary points of each segmented planar region (1 and 2); best fitted lines to each region boundary (3); cuboids created by projecting the 2D quad on the wall (4). [9]

Other works exists in building segmentation, that investigates the reconstruction of building facades and pipelines from point clouds as well as boundary identification and extraction.

Bosché et al. (2014) and Vosselman et al. (2004) investigate the reconstruction of facades and pipework from both ground scans and airborne LiDAR. They made use of region growing algorithms to categorize points that belong to facades of buildings. Additional constraints where imposed to correctly identify facades, such as size and position. The authors explain that there are several problems in the order in which the segmentation is made, data being over or under segmented as well as some objects being only partially segmented correctly. An important aspect is the hierarchy of the segmentation, some elements need to be segmented first to aid in the detection of others and some features can be identified only relating to other elements. For example, windows and doors are identified as openings in walls.

Thomson and Boehm (2016) constructed the model geometry by segmenting the point cloud in to vertical planes. On the vertical planes they extracted the boundaries and applied spatial reasoning to construct the needed geometry. They tested the algorithms on a wide variety of data sets, both interior and exterior with varying degrees of success. Later a spatial reasoning test is applied to merge planes, corresponding to walls, that are close together.

4.1.3. Boundary extraction methods

Boundary extraction is a crucial initial step before more complex tasks are performed on the data set, such as segmentation. Multiple attempts at automatically reconstruct object geometry from point clouds use a boundary detection algorithm (Rusu et al. 2008, Thomson and Boehm 2016) before extraction and/or recognition.

As segmentation and recognition, boundary detection methods are being developed in a multitude of domains, such as medicine, geomatics, computer science etc. Initially they where developed for optical character recognition and image analysis, and later they found their way in broader applications. In this sub-section a few boundary detection methods are presented.

Probably one of the first methods for boundary detection was presented by C. T. Zahn in 1966. In the paper “A FORMAL DESCRIPTION FOR TWO-DIMENSIONAL PATTERNS”, Zahn presents a method for obtaining a “structural description” of a dataset represented by a binary image. The “structural description”, here, consists of a set of polygonal curves arranged in a clockwise or counter-clockwise manner. These curves represent the continuous boundary between the connected points [17]. *“…selection of this particular format to describe the information content of a black/white pattern was motivated by a belief that the sequential trace of the boundary of an object contains the most useful data for recognizing objects one from another in a great many applications. Experimental evidence from the psychology of human visual perception and the neurophysiology of animal visual perception support this belief to an amazing extent.” [C. T. Zahn, 17]*

Since the inception of the method, many more have iterated and improved on it. Inspired by it, in 1986, Lunscher and Beddoes improve on the beforementioned method with a technique based on Euler numbers. Simply, a Euler number is defined as the difference between the total number of object in an image and the total number of holes in the respective image. They used the change in the Euler number to indicate the detection of a new closed object or hole in the image. Overall, for implementation of this method a modified Zahn curvature point detector is used, and the closure of a border is made by a Euler number closure detection system. [18]

New approaches to boundary detection employ different properties of the data. Canaz et al. (2015) make use of LiDAR data to automatically extract the boundary of water bodies. They present two algorithms for boundary detection. For the first, the centroid of the region of interest is computed and an imaginary line is generated between the centroid and the farthest point. Then, more imaginary lines are generated between the centroid and each point in the ROI. The angle between the main line and each generated line is computed to create a range between 0° and 360°, then for each degree the furthest point from the centroid is assigned as a boundary point. The second algorithm converts the cloud data into an image format. The resulting image is transformed into a binary representation and after that the pixels between the white and black colors are assigned as the object boundary. [19]

Li et al. (2013) fuse both imagery and LiDAR data to present an improved building boundary extraction method. They identify each roof patch from the LiDAR dataset firstly and secondly an improved Canny detector is used to extract the initial edges from the imagery data. Finally, the boundary is computed by fusing the roof patches with the initial boundary segmented from the images. [20]

4.1.4 Summary

Automatic object modelling from point cloud data spans across a wide variety of domains. Multiple attempts use different segmentation methods with varying results. It seems that the more promising methods employ a combination of segmentation methods and semantic information, with a boundary extraction component. The boundary extraction component may be used to aid in segmentation of objects with characteristic shape where segmentation methods based on local descriptors may exceed object boundary resulting in improper segmentation.

Chapter 5 – Modeling approach

5. Modelling approach

The modelling process involves 3 major steps. First, a planar segmentation is made. Next, on the extracted plane the boundary is computed, flowed by the creation of the surface. Throughout these steps, some filters are applied to clean the model and fit the data. The next subsections explain these processes in more detail.

The used programming environment for the development of this modelling approach is MatLab version R2018a.

Two scripts are created:

- *extract_rlv.m*
- *build_solid.m*

Two functions from the **Matlab Library** where used in the modelling process, each with explanation and references in their respective sections. These are:

- *find_delaunay_boundary03(X,Fd)*
- *writedxf(fname,X,Y,Z)*

5.1. Planar segmentation (extract_rlv.m)

Just as the direction of the X and Y axis can be used to identify vertical surfaces, the direction of Z axis can be used to identify and extract horizontal surfaces. This is of interest to us because the aim of this step is to extract a section from the cloud that can be used to create an approximated 2D floor plan drawing. In figure 31 below a workflow diagram of this step is presented.

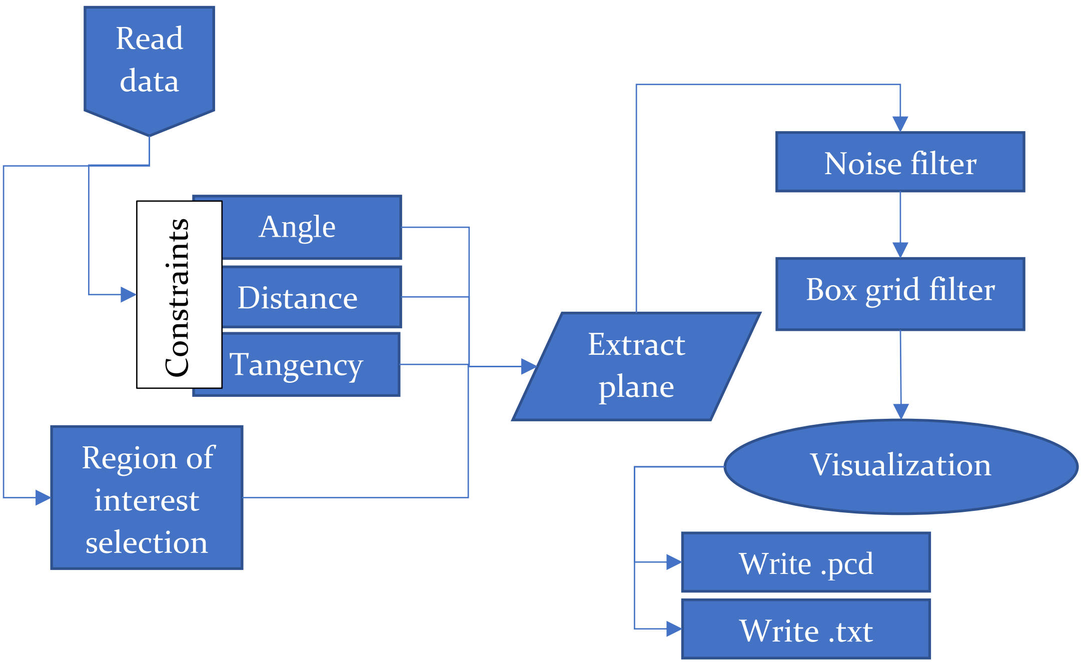


Figure 42. Workflow diagram of the planar segmentation step (extract_rlv.m)

The first step is to load the data. With the data loaded in, the process of setting the constraints and region of interest can begin. The constraints are needed to specify which plane needs to be extracted, here the Z axis is used as a constraint, forcing the planar segmentation to consider only planes that are on the direction of the vertical axis. Angle and distance constraints are also set here. The former two refer to the angular threshold and distance threshold in which the points need to be for the segmentation to perform as needed. The angular threshold refers to the maximal angular deviance for the coefficients of the found plane in relation to a perfect, horizontal in this case, plane. Figure 43 depicts how planes are extracted for different angle thresholds. On the other hand, the distance threshold is used to compare the residual of an observation to it, determining if the observation is an outlier or inlier. In this case a higher threshold means a thicker vertical plane extracted.

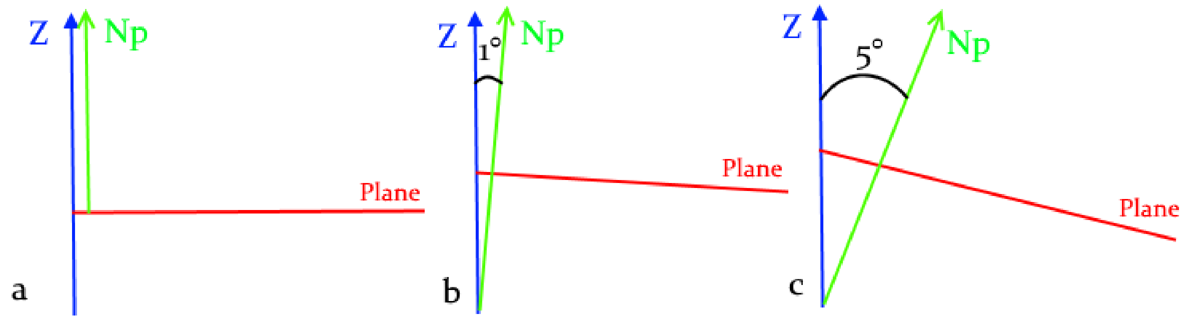


Figure 43. Planar extraction differences by angular threshold. a) 0° threshold - plane normal (green) and Z direction (blue) coincide; b) 1° threshold - the plane normal (green) can have a +/- 1° deviance from the Z direction (blue); c) 5° threshold – the plane normal (green) can have a +/- 5° deviation from the Z direction (blue).

Region of interest selection plays a role in the extraction for it allows to search for the best region in which the extracted data will better suit our needs. In this case the directions of the main axes (XYZ) are used to select a region best suited for the extraction. The extracted region will have as few outliers as possible and objects that are not of interest can be avoided (eg. furniture, light fixtures, etc.). With this we can also avoid holes in the data that appear due to improper data point acquisition by the scanner in regions with high clutter.

With the constraints in place and region of interest selected, a RANSAC algorithm is run on the data for the computation of planar coefficients and extraction of the best candidate. In this case the MatLab already existing implementation is used.

"Random sample consensus, or RANSAC, is an iterative method for estimating a mathematical model from a data set that contains outliers. The RANSAC algorithm works by identifying the outliers in a data set and estimating the desired model using data that does not contain outliers.

RANSAC is accomplished with the following steps

1. *Randomly selecting a subset of the data set*
2. *Fitting a model to the selected subset*
3. *Determining the number of outliers*
4. *Repeating steps 1-3 for a prescribed number of iterations"* [11]

On the extracted plane, two filters are run. A noise filter and a box grid filter. The purpose of these filters is to smooth the data for the following steps of the process. Here, also already existing MatLab implementations where used.

For the noise filter "pcdenoise" is used. It is based on Rusu et al. (2008). This filter is a sparse outlier removal method that computes the mean μ and standard deviation σ of nearest neighbor distances and eliminates the points that which fall outside the $\mu \pm \alpha \cdot \sigma$ interval. The value of α changes depending on the size of the analyzed neighborhood. [12]

The box grid filter is based on Pomerleau et al. (2013). Here the gridAverage downsample methods is used. It merges all the data from the same cube in one point by computing the axis-aligned bounding box of the entire point cloud and then dividing it in grid boxes of the same size, specified by the user. Color and normal information are also averaged. This filtering method was used in favor of other methods due to its nature to better preserve the original form of the point cloud.

At the end of this step figures of the extracted plane (Fig. 44a), filtered plane (Fig. 44b) and remaining data (Fig. 44c) will be displayed to screen. Further, both a .pcd and a .txt file containing the points associated with the extracted plane are saved to disk. These files are used in the next steps for the modelling.

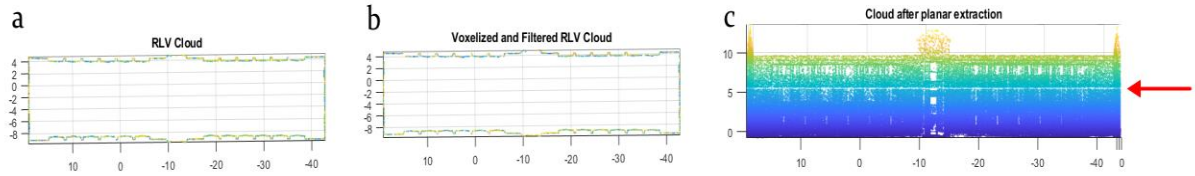


Figure 44. Figures output from the matlab script 'extract_rlv.m'. The red arrow to the left shows from where the plane from 'a' was extracted.

5.2. Solid object creation (build_solid.m)

In this part, the files generated at the previous steps are used to build a solid model. The previously extracted planar section from the point cloud is now loaded and the first parameter set. Here the first parameter is d_{max} which represents the maximum point-to-point distance in which the points will be considered part of the same entity. Further, a boundary extraction function is performed, followed by a smoothing filter, and finally, the model is generated. Figure 34 below shows a diagram flowchart of this part of the modelling approach.

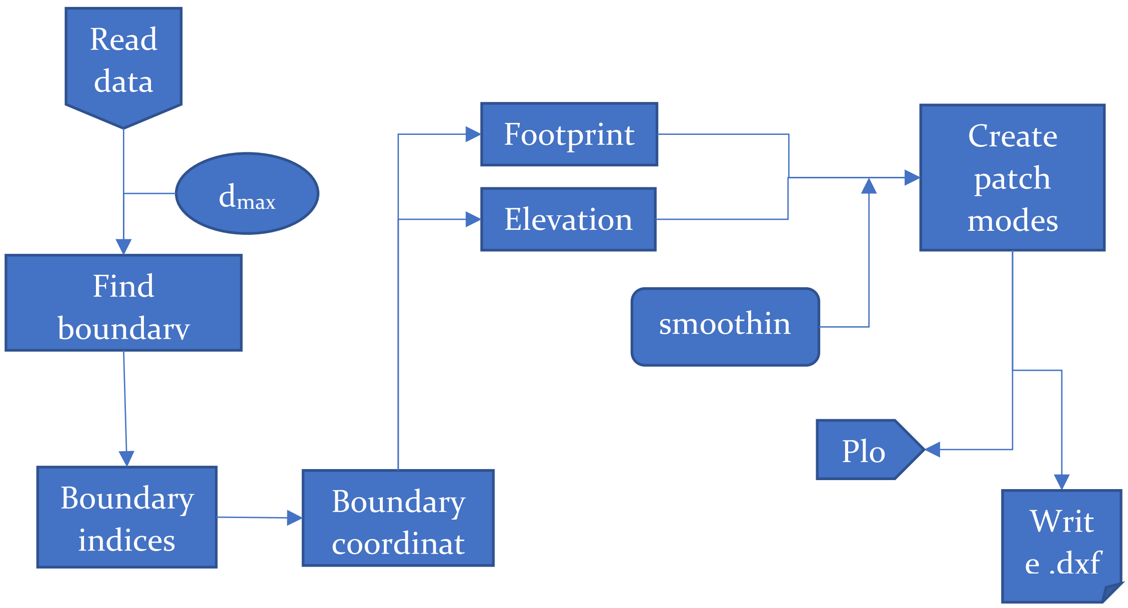


Figure 45. Workflow diagram of the solid object creation step (build_solid.pcd)

With the data from the previous step loaded (4.1 Planar segmentation) and d_{max} set, the boundary finding function '*find_delaunay_boundary03(X,Fd)* ' is used.

find_delaunay_boundary03(X,Fd) is a function developed by M. Awrangjeb (2016). Its role is to extract the outline of rectilinear buildings from a set of points. An edge is the line that connects two consecutive points on the boundary. Then, they are traced to form an irregular building outline.

The function is composed of two main steps: boundary identification and boundary tracing. In the first step, the points at the periphery of the shape (building) are connected to for an unordered set of edges. In the second step, the unordered edges found previously are tracked in order to find an ordered set of points. This function also works for multiple disconnected objects in the same data set, if point-to-point distance between the objects is at least double the distance between the points in the objects themselves. [15]

To create the solid model, the indices of the extracted boundary are converted to point coordinates and a filled polygon, representing the building footprint, is generated. Further, elevation needs to be created. In this case the height of the building was measured and because the building elements are symmetric, vertically and horizontally, the footprint polygon is offset in the Z axis direction by the desired (measured/calculated) height. In this section, a smoothing filter is introduced to smoothen the boundary even further. The used filter is the 1-D median filter already implemented in MatLab. The filter applies a third-order one-dimensional median filter to the input and is based on "Pratt, William K. Digital Image Processing. 4th Ed. Hoboken, NJ: John Wiley & Sons, 2007". [16]

The final step is constructing the object geometry. For this the two polygons created earlier are taken in account. For each point on the polygon boundary, the corresponding point on the other polygon's boundary is identified and a line is computed between the two. This process is repeated for each point that is contained in the boundary, resulting in a model composed only of lines. Then, the area between each two consecutive lines is filled to create the solid object.

In the last part, a plot is made to visually inspect the output and the generated model is saved to disk in .dxf format so further operations can be made in a CAD program. The conversion to .dxf is made using the function *writedxf(fname,X,Y,Z)*.

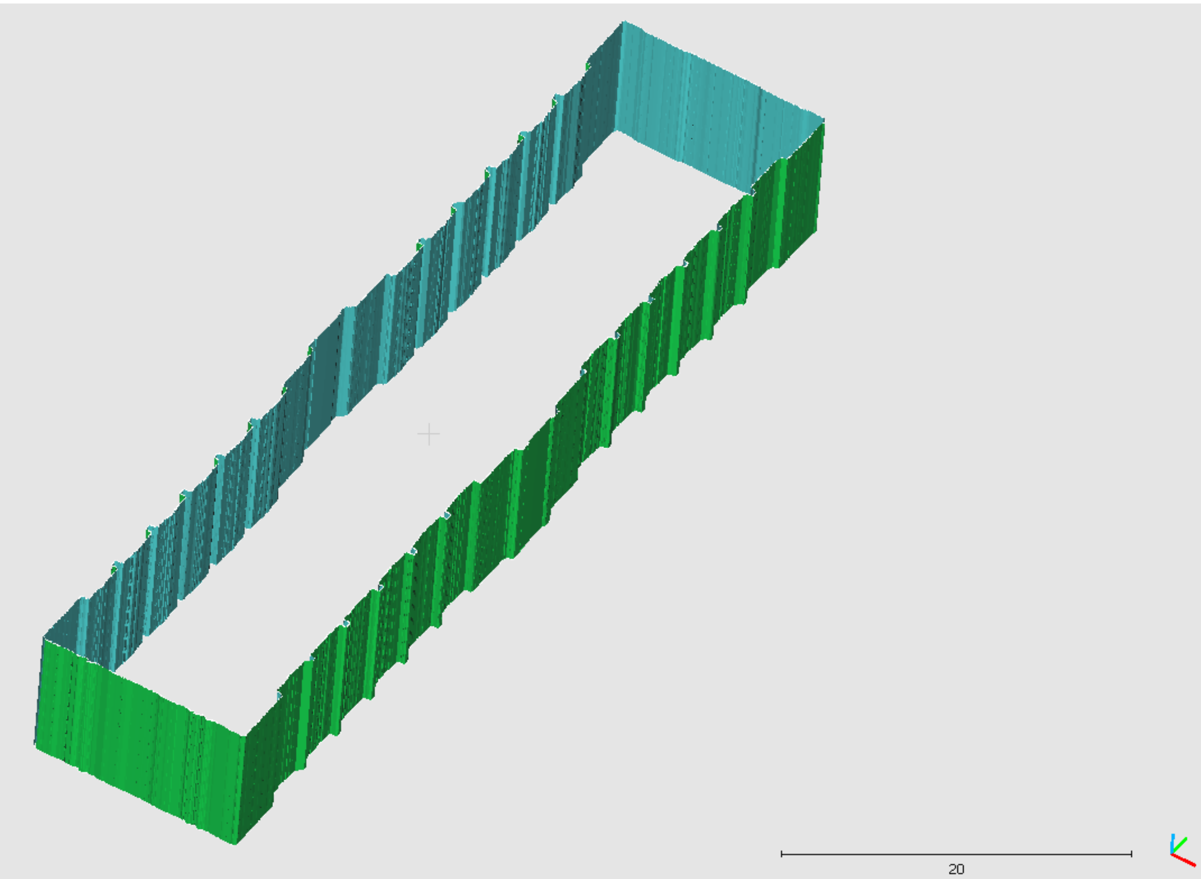


Figure 46. Final solid model. Green - wall exterior, cyan - wall interior

writedxf(fname,X,Y,Z) was developed by Greg Siegle. It writes a dxf file, given a file name and a 3D mesh specified in XYZ. [21]

Chapter 6 – Accuracy evaluation

The last chapter of this paper will evaluate the accuracy of the point cloud and the model compared against the survey done with the total station. To do so, all sides of the building were measured, on the point could, on the model and on the points resulted from the total station survey resulting the table in Appendix B:

The first column of the table represents the orientation of the measured building wall in 2D. this will be useful later in this chapter when we will check if this factor influences the accuracy of our data.

The third column represents the measurements done on the raw point cloud. This was the most difficult and time consuming part, due to the fact that it is impossible to do on the raw point cloud. The main issues being that there are too many points and because of that it is impossible to select two points with the same height so the distance measurement will not be altered. To solve this problem, a slice of the point cloud was sectioned and then dropped on the 2D plane therefore eliminating the height issue and making the point cloud manageable.

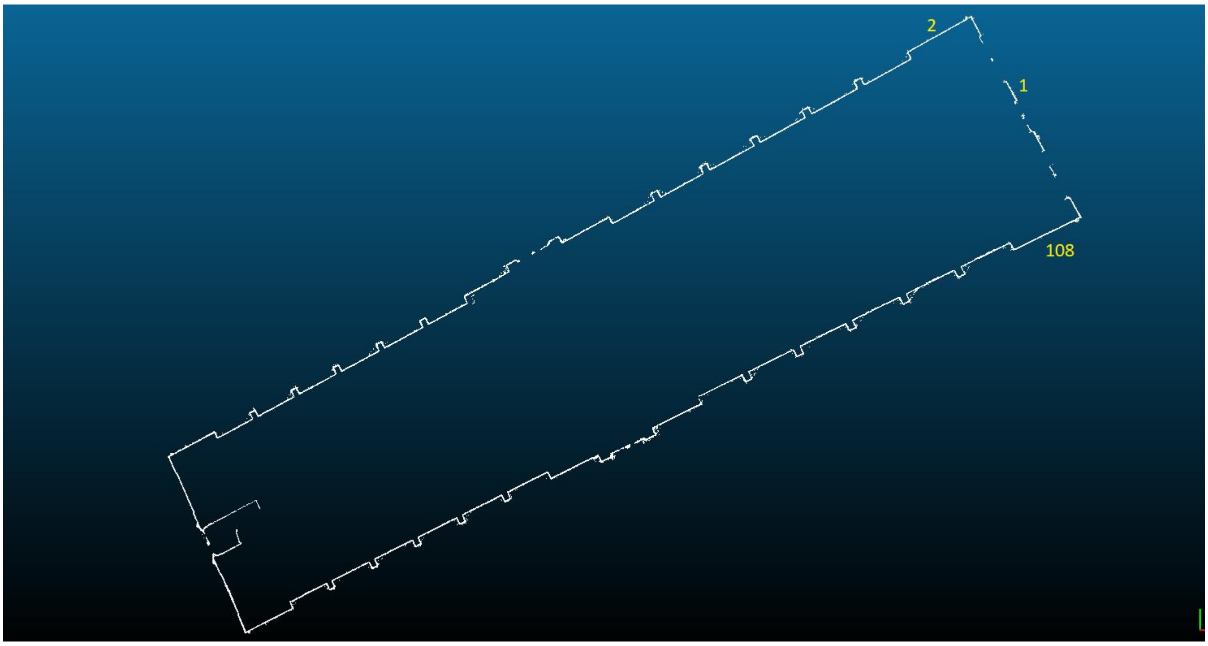


Figure 48 2D representation of the exterior of the building

The figure above shows the 2D footprint of the building which was used for the distance measurements on the point cloud. The first distance measured was the NE side of the building facing the freljord, followed by all sides of the building in a counter clock wise order. The distances were measured by manually selecting the corners of each segment and then computing the distance between the two points using Cloud Compare.

The last two columns, representing the distances measured on the total station survey and on the model, were measured in AutoCAD.

Further the distances measured on the point cloud and model will be compared to the total station survey in order to check the accuracy of the point cloud and the model.

The comparison consists of analyzing three data sets:

- Delta which represents the distance difference between the total station survey and the point cloud.
- Delta2 which represents the distance difference between the total station survey and the model.
- Delta3 which represents the distance difference between the point cloud and the model.

We expect to have differences raging between 3 cm and 10 cm because the scanner used to collect the data has an accuracy of about 3 cm.

Further, the standard deviation of each Delta will be computed, and then a check will be made to see if the values of the Deltas are normally distributed.

Standard deviation.

The standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation means that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values.

The standard deviation of a data se is equal to the square root of its variance and it is commonly used to measure confidence in statistical conclusions. One useful property of the standard variation is that, unlike variance, it is expressed in the same units as the data.

$$\sigma = \sqrt{\frac{\sum d_i^2}{n}}$$

σ– standard deviation

d_i^2 – represents the multiplication of vector d with its transpose.

n – number of elements in vector d

Normal Distribution

In probability theory, the normal distribution is a very common probability distribution. It is important in statistics and are usually used in natural or social sciences to represent random variables whose distributions are not known.

The normal distribution is useful because of the central limit theorem, which states that averages of samples of observations of random variables independently drawn from independent distributions converge in distribution to the normal, that is, become normally distributed when the number of observations is sufficiently large.

To compute the standard deviation for our three data sets (Delta, Delta2, Delta3), a MatLAB script was made. It has as input data one of our three data sets, which we will treat individually and as output, two bar charts showing the standard deviation and normal distribution of the said data set.

6.1 Standard deviation and normal distribution of Delta

As stated above, our first data set, Delta represents the distance differences between the scanned survey, and the total station survey.

The first step is to compute the mean value of this data set. The mean value represents the sum of all numbers in the data set divided by the number of elements.

$$\mu = \frac{d_1 + d_2 + d_3 + \cdots + d_n}{n}$$

d – values from the data set

n – number of values in the data set

μ – mean value of the data set

In this case the mean value of our data set is equal to μ = 0.000918. This is to be expected since all our values are smaller than 1.

This small value indicates that the values from our data set are most likely very close to the mean.

Last but not least, we will discuss about the standard deviation of this data set and look at the graphs which resulted from the MatLAB script.

The standard deviation value in this case is σ = 40.9 mm. This means that all the values from our data set are close to the mean value. This can also be observed in the graph below.

As expected, the graph shows that the values converge near the mean value of the data set with the number of values increasing with each interval until the middle then decreasing towards the end resembling a bell shape.

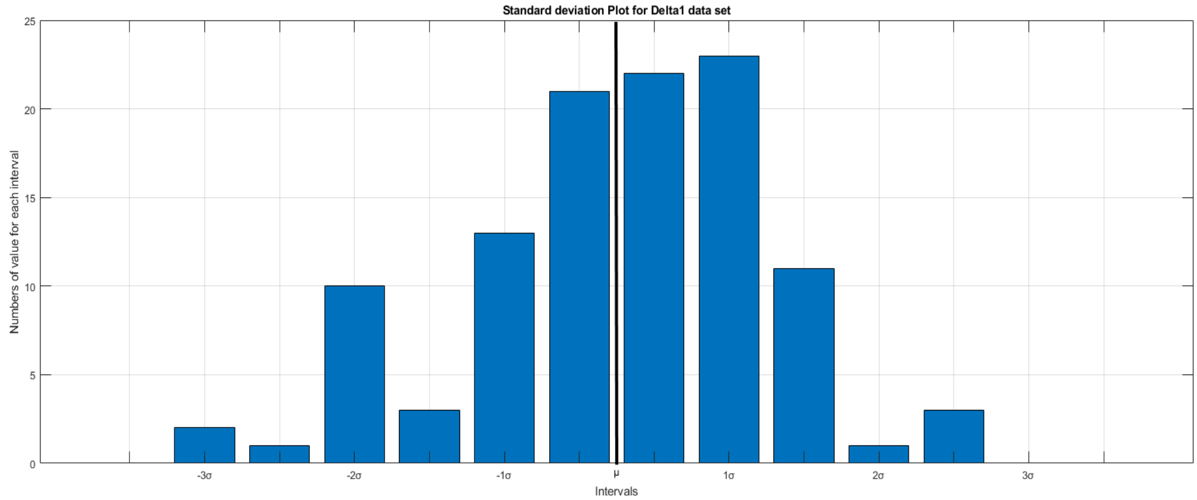


Figure 49 Standard deviation plot of the Delta data set

Another important factor is that, because the data tends to be around the central values with no bias left or right, meaning that the values are normally distributed. The "bell curve" is a normal distribution and the histogram above shows that data follows closely but not perfectly which is to be expected.

The 68-95-99.7 rule.

In statistics, the 68-95-99.7 rule is a shorthand used to remember the percentage of values that lie within a band around the mean in a normal distribution with a width of two, four and six standard deviations. This means that 68% of the values lie within one standard deviation away from the mean, 95% lie within two standard deviation away from the mean and 99.73% of the values are three standard deviations away from the mean.

In our case the data set represents 110 values, 68% of which is approximately 75.

In our case the number of values which are within the [-1σ, 1σ] is 79, representing 71.81% of our total values.

The next step is to check if 95% of the values are within two standard deviations away from the mean. 95% of 110 is approximately 104, so there should be around 105 numbers within this interval.

When we add up the numbers the result amounts to 104, which represents 94.54% of our total values.

Last, we will check if the number of values within three standard deviations from the mean represent 99.73% of the total values. 99.73% of 110 is approximately 109. When we add up the number of values within this interval, the result is 110 representing 100%.

In conclusion even if we did not obtain perfect results to the 68-95-99.7 rule, the values are close enough to the desired value, therefore we can safely say that Delta is a normal distributed data set.

6.2 Standard deviation and normal distribution of the Delta2 data set

For the second data set, Delta2, which represents the distance measurements differences between the total station survey and the model, we will follow the same steps as for the first data set.

Same as for our first data set, the first step is to compute the mean value and standard deviation, which in this case are μ=-0.0119 and σ = 23.9 mm.

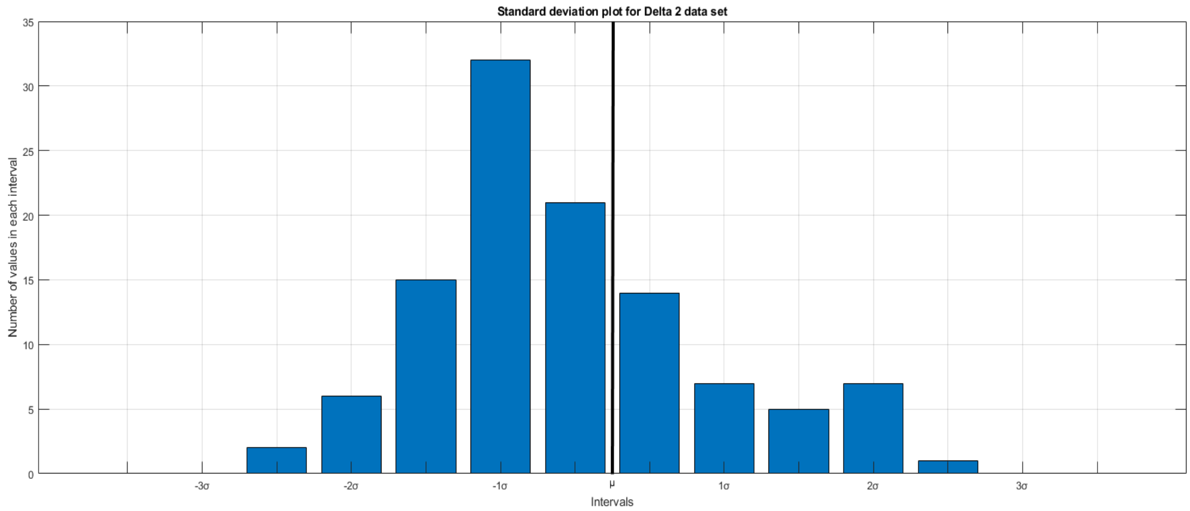


Figure 50 Standard deviation plot for Delta2 data set

As we can observe in this graph the main bulk of the values are not the closest to the mean, but they are still within the [-1σ,1σ] interval.

The next step is to see if the data set meets the requirements of the 68-95-99.7 rule.

In this case the first interval, [-1σ,1σ], consists of 74 values representing 67.27% of total values. The second, [-2σ,2σ], contains 107 values representing 97.27% of total values and the third consists of 110 values representing 100%. This means that no values stray more that 3σ away from the mean.

Because only the first condition was off by a mere 0.73% and because no values stray further away than 3σ, this data set is normally distributed.

6.3 Standard deviation and normal distribution of the Delta3 data set

The third data set, Delta3, which represents the distance difference values between the point cloud and the model will be tested in the same way as the previous two. The mean value of Delta3 is μ=-0.011, similar to the previous data set.

The standard deviation of this third data set is σ=41.0 mm a similar value to the standard deviation of our first data set, Delta. Because of this we expect that most values from this data set to be close to the mean value.

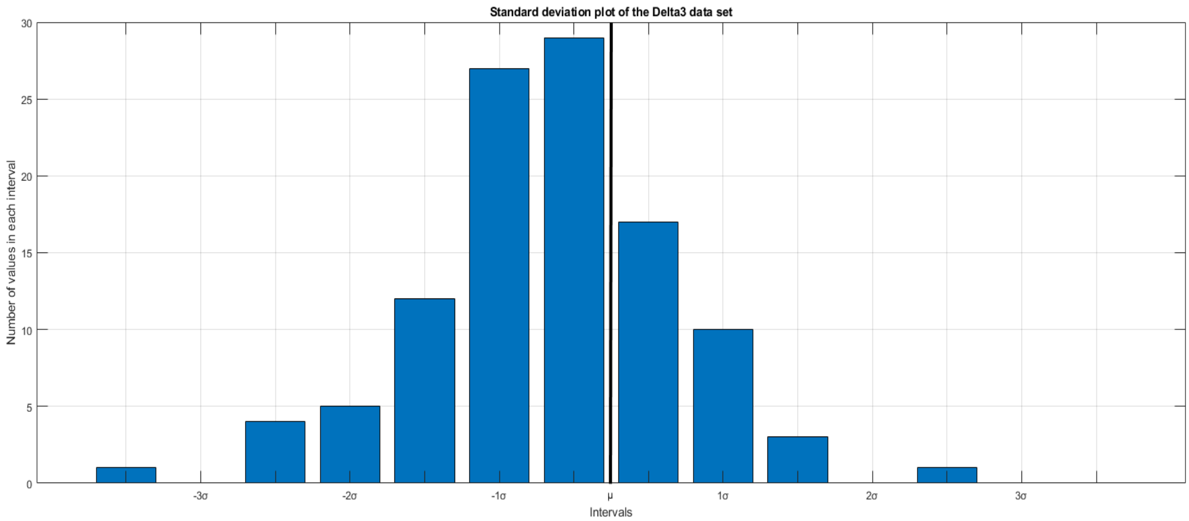


Figure 53 Standard deviation plot for Delta3 data set

Figure 53 represents the standard deviation plot for the Delta3 data set. The graph, same as the one from the previous data set, still closely but not perfectly follows a "bell shape" curve which led us to believe that this data set is also a normal distribution.

Further we will test the 68-95-99.7 rule for this data set.

For the first condition to be true from the total of 110 values of the data set, 75 of them should be within 1 standard deviation from the mean.

When we sum the number of values within this interval the result is 83 representing 75.45% of the total number of values.

For the second condition, the total number of values within the [-2 σ,2 σ] interval should represent 95% of the total values from this data set.

When we sum up the amount of values within this interval, the result is 103 representing 93.63% of our total values

Finally, the last interval [-3 σ, 3 σ] should contain 99.73% of the total values, but in this case, it contains 108 values representing 98.18%.

Because the second and third condition were not fulfilled, but the result are close to the desired value, being only about 1.5% off, we can assume that this data set is approximately normally distributed.

Figure 53 shows a histogram of values from the Delta3 data set. From this graph we can spot at least 3 values that stray too much from the mean.

Because these three values deviate too much from the mean, even though they are not consecutive, we can consider these values as anomalies.

Figure 51 and Figure 52 show the histograms of values from the Delta and Delta 2 data sets. When compared to Figure 53 we can observe that we had a normally distributed data sets values don't stray that much from the mean and are more evenly distributed.

Another important observation is that Delta3 is the only data set that has some of its values stray more than 3σ away from the mean.

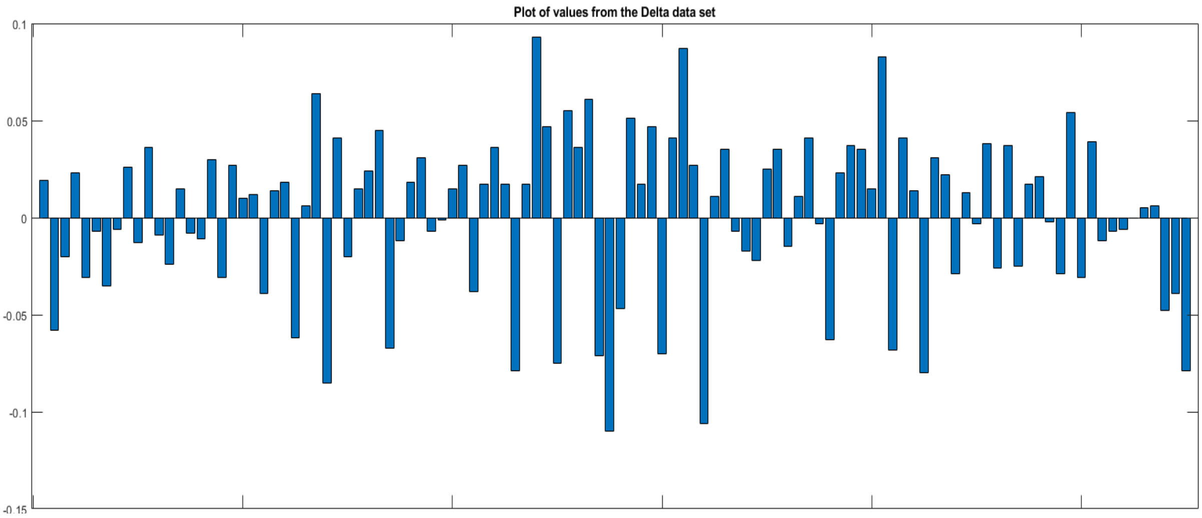


Figure 51 Plot of values from the Delta data set

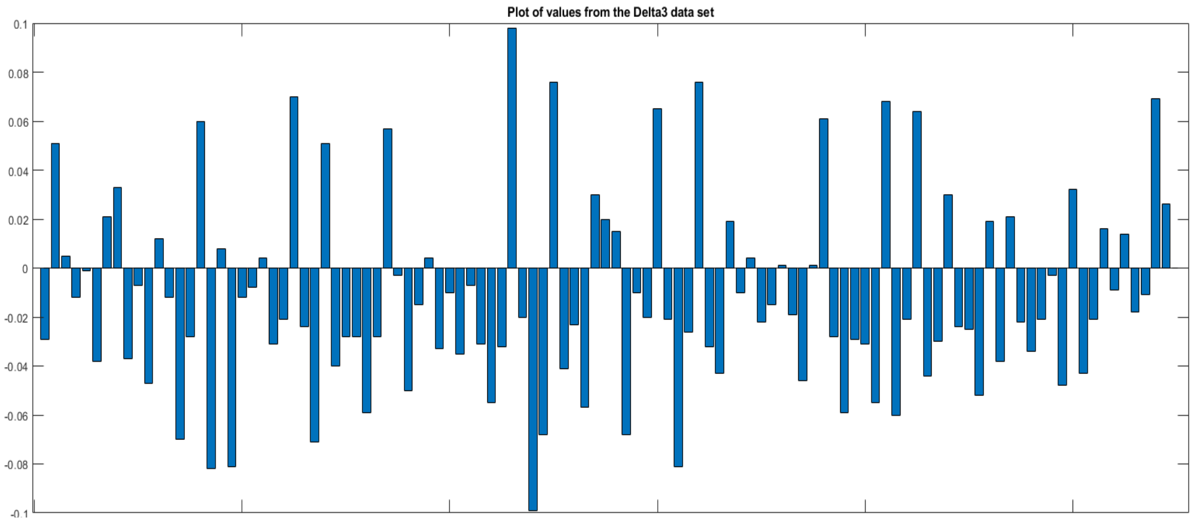


Figure 52 Plot of values from the Delta2 data set

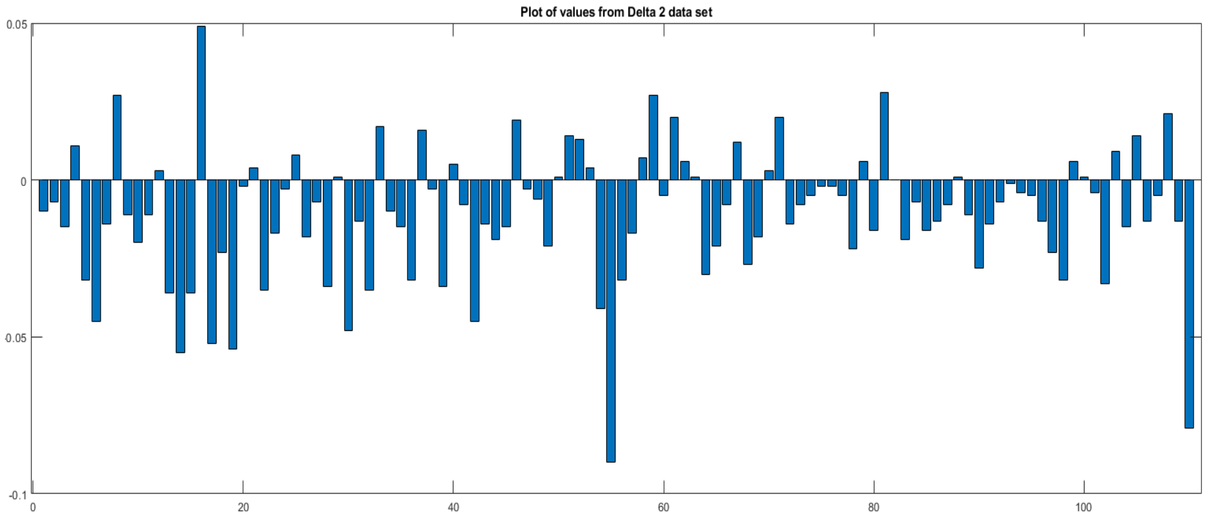


Figure 53 Plot of values from the Delta3 data set

6.4 Standard deviation and normal distribution of Delta4 data set.

Given the shape and symmetry of the building, we wanted to know if the orientation of the walls has any impact on our data. We divided each previous data set in two, based on the alignment of the walls and we will compare the distances differences computed for the walls parallel to the Y axis with the distances differences computed for the walls parallel to the X axis.

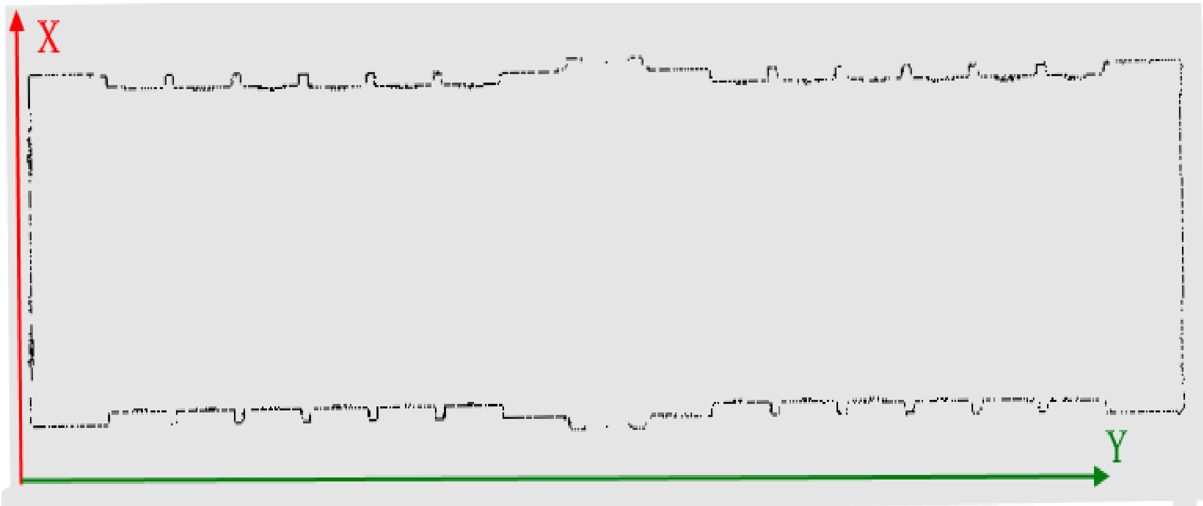


Figure 54 Walls orientation on XY axis

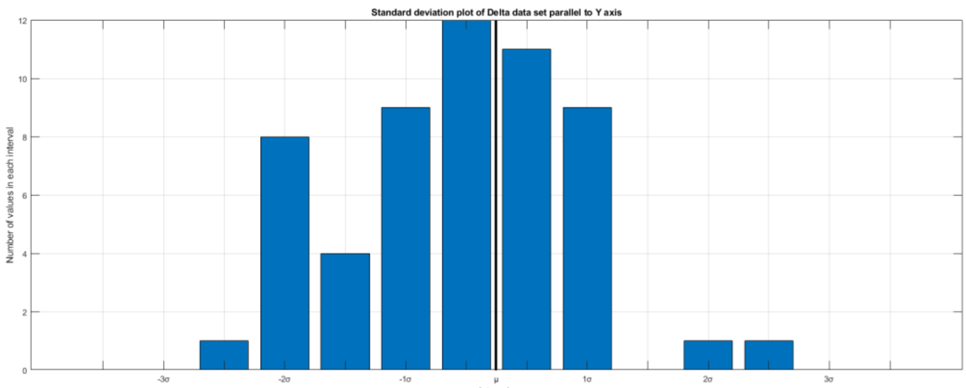


Figure 55 Standard deviation plot for Delta parallel to Y axis

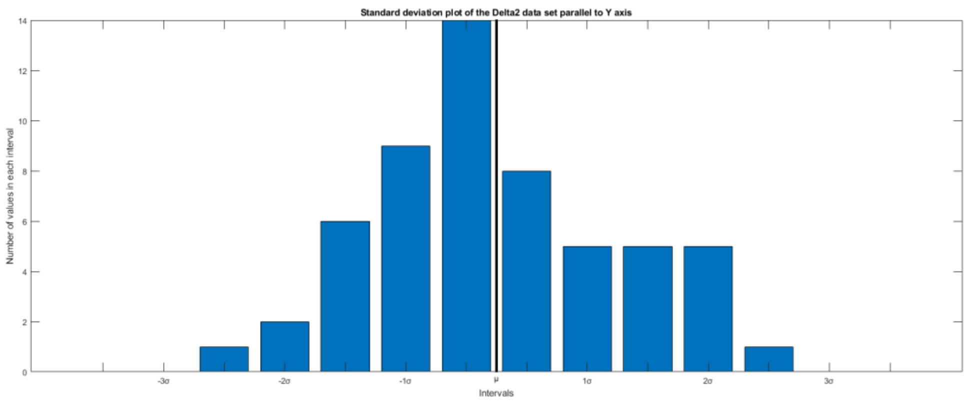


Figure 56 Standard deviation plot for Delta2 parallel to Y axis

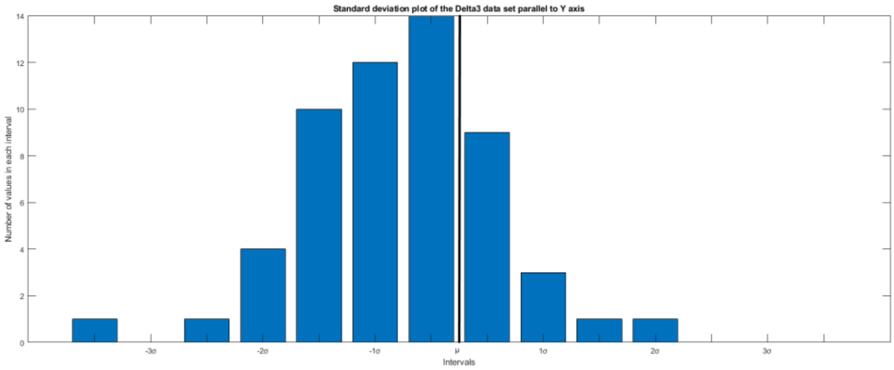


Figure 57 Standard deviation plot for Delta3 parallel to Y axis

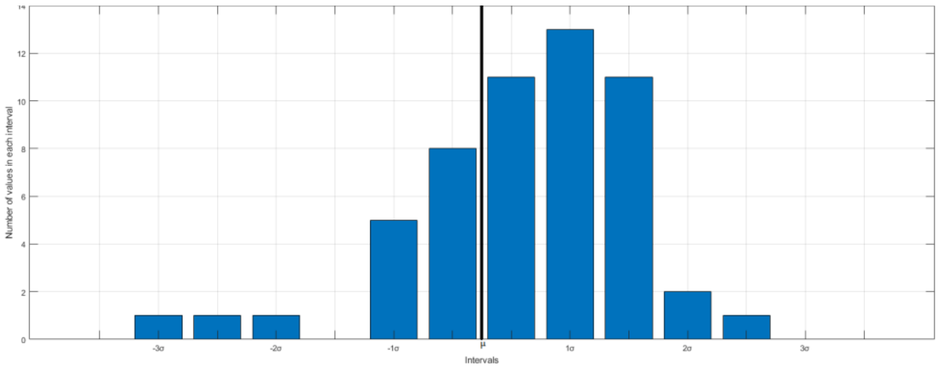


Figure 58 Standard deviation plot for Delta parallel to X axis

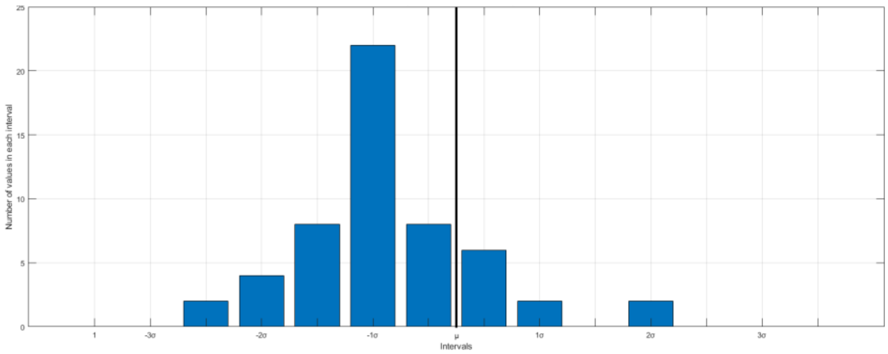


Figure 59 Standard deviation plot for Delta2 parallel to X axis

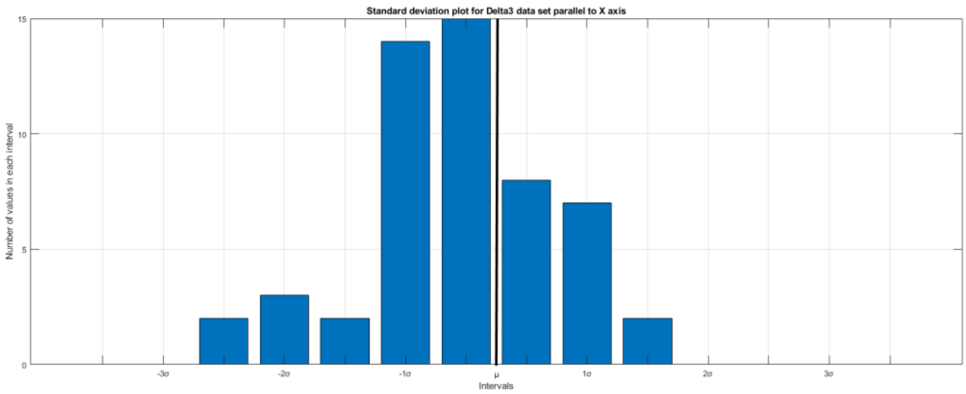


Figure 60 Standard deviation plot for Delta3 parallel to X axis

	Standard deviation (mm)	Mean
Delta || X	37.8	0.014
Delta2 || X	22.4	-0.023
Delta3 || X	34.4	-0.008
Delta || Y	43.6	-0.012
Delta2 || Y	43.3	-0.0004
Delta3 || Y	25.2	-0.013

Figure 61 Table containing the standard deviation and mean value for Figure 55-60

We can observe from the standard deviation plots that there is no major difference when it comes to normal distribution of the data. All plots have a "bell shape" meaning that they are normally distributed. The only data set that has values outside the [-3σ, 3σ] is Delta3 same as in our previous experiment. An interesting observation is that the 2 anomalies that exist, are both values that are part of the data set that is parallel to the Y axis. There are no anomalies on the data set that is parallel to the X axis.

Chapter 7 Conclusion

1. What are current methods to automatically create a solid object from point cloud data?

Even though the idea of an automatically workflow that transforms point cloud data into a useable model is not new, there are no fully automatic solutions developed yet. Some big developers like Autodesk, Leica, ImaginIT developed some semi-automated solutions that target specific parts of this process such as automated wall detection, automated surface detection or semi-automated pipe detection, but no developer has built a complete solution that encompasses all these elements.

Research in this field has started in the late 20th century, in the field of computer science, but until recently this technology was too expensive and too hard to deal with due to the large amount of collected data and required processing power.

2. What are the steps in developing a model from cloud data?

The modelling process consists of three major steps: planar segmentation, boundary computation on the extracted plane and lastly, surface creation. This process is far from being the final solution to the problem, but can be starting point from which a solution that eliminates the gap between data collection and data processing can be found.

3. How accurate is the model?

When assessing the model's accuracy 110 distances were measured on the model and then compared with the total station survey. It was determined that the data set was approximately normally distributed with a mean $\mu = 0.011$ and a standard deviation $\sigma = 41$ mm, meaning there were some anomalies present. These anomalies most likely appeared because of tempering with the data during the modelling process (filters) and because determining the corners of the building was particularly difficult due to the nature of the point cloud data.

Because the data set was approximately normal distributed and because the standard deviation was a small value we assume that there were no systematical errors.

Future prospects

Based on what was learned until now, for the development of this project, some ideas on how to improve this process can be formulated.

Improvements can be made in the boundary extraction step within sub-section 4.2. A corner detection step can be implemented in such a way that each wall is extracted as a separate entity with its extremities computed and saved. This will allow for each individual wall to be indexed into a database, with its associated dimensions. Indexing each wall as its own entity can also play a role in accuracy assessments. For example, in concordance with the presented accuracy assessment method the orthogonality or parallelism can be tested.

Future development can be made in adapting the code to different, more complex data sets. For this, multiple planes can be extracted, using the same process as in sub-section 4.1. then the model can be generated as presented in sub section 4.2.

Data capture technologies evolve at a rapid pace, and the scanner used in this project have updated versions. Mobile scanning systems are more complex than static scanners thus harder to understand, leading to errors when improper calibrated. Therefore, better instrumentation will lead to better data capture and model generation. Instrumentation that can capture materials to add to the existing geometric information would be hugely beneficial. However, this is extremely challenging due to the different representation materials have.

Sources

[1]https://web.archive.org/web/20141016190503/http://www.nationalbimstandard.org/faq.php#faq1

Archived from the original on 16 October 2014.

[2]https://www.arcdox.com/copy-of-what-is-bim

[3]https://geoslam.com/

[4] GeoSLAM Desktop Processing Software Guide ©, GeoSLAM Ltd., Version 3.2.0., 2017

[5]https://www.engineering.com/BIM/ArticleID/11436/BIM-101-What-is-Building-Information-Modeling.aspx

[6]https://www.gsa.gov/real-estate/design-construction/3d4d-building-information-modeling/bim-guides/bim-guide-01-bim-overview

[7]https://shop.bsigroup.com/forms/PASs/PAS-1192-2/

[8] https://www.youtube.com/watch?v=DcjHGc3Z24w&t=352s * images taken from the video

[9] R.B. Rusu et al. "Robotics and Autonomous Systems", 2008, pp. 927–941

[10] Charles Patrick Hugo Thomson, "From Point Cloud to Building Information Model - Capturing and Processing Survey Data Towards Automation for High Quality 3D Models to Aid a BIM Process", 2016

[11] https://se.mathworks.com/discovery/ransac.html

[12] Rusu, R. B., Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. "Towards 3D Point Cloud Based Object Maps for Household Environments", Robotics and Autonomous Systems Journal, 2008.

[13] Pomerleau, F., F. Colas, R. Siegwart, and S. Magnenat. "Comparing ICP variants on real-world data sets." Autonomous Robots. Vol. 34, Issue 3, April 2013, pp. 133–148.

[14] https://se.mathworks.com/help/vision/ref/pcdownsample.html#bupqqn1-1-gridAverage

[15] M. Awrangjeb, "Using point cloud data to identify, trace, and regularize the outlines of buildings" International Journal of Remote Sensing, Volume 37, Issue 3, February 2016, pages 551-579. Open access at: http://www.tandfonline.com/doi/pdf/10.1080/01431161.2015.1131868

[16] https://se.mathworks.com/help/signal/ref/medfilt1.html

[**17**] C. T. Zahn, “A FORMAL DESCRIPTION FOR TWO-DIMENSIONAL PATTERNS”, Stanford Linear Accelerator Center Stanford University, Stanford, California, 1966, pp. 621-628.

[**18**] W. H. H. J. LUNSCHER, M. P. BEDDOES, “Fast Binary-Image Boundary Extraction”, COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 38, 1987, pp. 229-257.

[**19**] Sibel Canaz, Fevzi Karsli, Abdulaziz Guneroglu, Mustafa Dihkan, “Automatic boundary extraction of inland water bodies using LiDAR data”, Ocean & Coastal Management 118, 2015, pp. 158-166

[**20**] Yong Li, Huayi Wub, Ru Ana, Hanwei Xua, Qisheng Hea, Jia Xua, “An improved building boundary extraction algorithm based on fusion of optical imagery and LIDAR data”, Optik 124, 2013, pp. 5357–5362

[**21**] https://se.mathworks.com/matlabcentral/fileexchange/333-writedxf-m

[22] Dr. Mostafa Abdel-Bary Ebrahim “3D Laser Scanners: Hystory, Applications and future”, October 2011.

Appendix A – Digital upload contents

Below is an overview of the data uploaded with this project. The upload contains the source code used in the routines described in Chapter 4, the data on which they were tested and the output.

- output_exterior contains the output of the routines from Chapter 4;
- test_data contains the initial data set on which the processing from Chapter 4 is made;
- below the folders, the scripts used are provided.

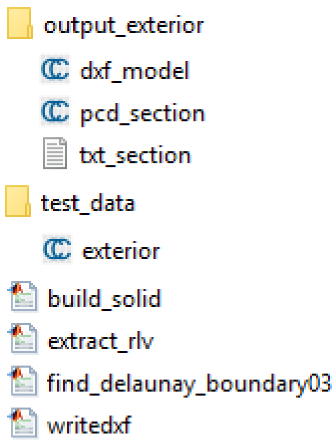


Figure A.1. Structure of uploaded data

Appendix B – Distance measurements

All sides of the building were measured, on the point could, on the model and on the points resulted from the total station survey resulting the following table:

Table B.1. Table containing the distance measurements

Exterior				
Orientation in 2D	Nr.	Point cloud(m)	Total Station(m)	Model(m)
--	1	13.709	13.69	13.719
|	2	4.193	4.251	4.2
--	3	0.455	0.475	0.47
|	4	3.139	3.116	3.128
--	5	0.445	0.476	0.477
|	6	0.443	0.45	0.488
--	7	0.457	0.492	0.471
|	8	3.127	3.133	3.1
--	9	0.457	0.431	0.468
|	10	0.463	0.476	0.483
--	11	0.466	0.43	0.477
|	12	3.134	3.143	3.131
--	13	0.446	0.47	0.482
|	14	0.438	0.423	0.493
--	15	0.453	0.461	0.489
|	16	3.145	3.156	3.096
--	17	0.467	0.437	0.519
|	18	0.453	0.484	0.476
--	19	0.464	0.437	0.518
|	20	3.125	3.115	3.127
--	21	0.468	0.456	0.464
|	22	0.448	0.487	0.483
--	23	0.47	0.456	0.487
|	24	3.105	3.087	3.108
--	25	0.476	0.538	0.468
|	26	3.586	3.58	3.604
--	27	0.477	0.413	0.484
|	28	4.042	4.127	4.076
--	29	0.461	0.42	0.46
|	30	3.556	3.576	3.604

--	31	0.47	0.455	0.483
|	32	3.099	3.075	3.134
--	33	0.5	0.455	0.483
|	34	0.454	0.521	0.464
--	35	0.468	0.48	0.483
|	36	3.111	3.093	3.143
--	37	0.495	0.464	0.479
|	38	0.465	0.472	0.468
--	39	0.456	0.457	0.49
|	40	3.125	3.11	3.12
--	41	0.488	0.461	0.496
|	42	0.453	0.491	0.498
--	43	0.474	0.457	0.488
|	44	3.115	3.079	3.134
--	45	0.474	0.457	0.489
|	46	0.475	0.554	0.456
--	47	0.474	0.457	0.477
|	48	3.147	3.054	3.153
--	49	0.474	0.427	0.495
|	50	0.451	0.526	0.45
--	51	0.482	0.427	0.468
|	52	3.144	3.108	3.131
--	53	0.483	0.422	0.479
|	54	4.163	4.234	4.204
--	55	13.665	13.775	13.755
|	56	4.187	4.234	4.219
--	57	0.473	0.422	0.49
|	58	3.125	3.108	3.118
--	59	0.474	0.427	0.447
|	60	0.456	0.526	0.461
--	61	0.468	0.427	0.448
|	62	3.141	3.054	3.135
--	63	0.484	0.457	0.483
|	64	0.448	0.554	0.478
--	65	0.468	0.457	0.489
|	66	3.114	3.079	3.122
--	67	0.484	0.491	0.472
|	68	0.444	0.461	0.471
--	69	0.456	0.478	0.474
|	70	3.135	3.11	3.132

--	71	0.492	0.457	0.472
|	72	0.457	0.472	0.471
--	73	0.475	0.464	0.483
|	74	3.134	3.093	3.139
--	75	0.477	0.48	0.479
|	76	0.458	0.521	0.46
--	77	0.478	0.455	0.483
|	78	3.112	3.075	3.134
--	79	0.49	0.455	0.484
|	80	3.591	3.576	3.607
--	81	0.503	0.42	0.475
|	82	4.059	4.127	4.059
--	83	0.454	0.413	0.473
|	84	3.594	3.58	3.601
--	85	0.458	0.538	0.474
|	86	3.118	3.087	3.131
--	87	0.478	0.456	0.486
|	88	0.458	0.487	0.457
--	89	0.469	0.456	0.48
|	90	3.112	3.115	3.14
--	91	0.475	0.437	0.489
|	92	0.458	0.484	0.465
--	93	0.474	0.437	0.475
|	94	3.131	3.156	3.135
--	95	0.478	0.461	0.483
|	96	0.444	0.423	0.457
--	97	0.468	0.47	0.491
|	98	3.114	3.143	3.146
--	99	0.484	0.43	0.478
|	100	0.445	0.476	0.444
--	101	0.47	0.431	0.474
|	102	3.121	3.133	3.154
--	103	0.485	0.492	0.476
|	104	0.444	0.45	0.459
--	105	0.476	0.476	0.462
|	106	3.121	3.116	3.134
--	107	0.481	0.475	0.486
|	108	4.203	4.251	4.182
|	109	61.906	61.945	61.919
|	110	61.866	61.945	61.945

Appendix C – Values in each dataset

Delta represents the distance difference between the total station survey and the point cloud.

Delta2 represents the distance difference between the total station survey and the model.

Delta3 represents the distance difference between the point cloud and the model.

Table C.1. Table representing the values in each data set

Delta(m)	Delta2(m)	Delta3(m)	Delta(m)	Delta2(m)	Delta3(m)
0.019	-0.029	-0.01	-0.047	0.015	-0.032
-0.058	0.051	-0.007	0.051	-0.068	-0.017
-0.02	0.005	-0.015	0.017	-0.01	0.007
0.023	-0.012	0.011	0.047	-0.02	0.027
-0.031	-0.001	-0.032	-0.07	0.065	-0.005
-0.007	-0.038	-0.045	0.041	-0.021	0.02
-0.035	0.021	-0.014	0.087	-0.081	0.006
-0.006	0.033	0.027	0.027	-0.026	0.001
0.026	-0.037	-0.011	-0.106	0.076	-0.03
-0.013	-0.007	-0.02	0.011	-0.032	-0.021
0.036	-0.047	-0.011	0.035	-0.043	-0.008
-0.009	0.012	0.003	-0.007	0.019	0.012
-0.024	-0.012	-0.036	-0.017	-0.01	-0.027
0.015	-0.07	-0.055	-0.022	0.004	-0.018
-0.008	-0.028	-0.036	0.025	-0.022	0.003
-0.011	0.06	0.049	0.035	-0.015	0.02
0.03	-0.082	-0.052	-0.015	0.001	-0.014
-0.031	0.008	-0.023	0.011	-0.019	-0.008
0.027	-0.081	-0.054	0.041	-0.046	-0.005
0.01	-0.012	-0.002	-0.003	0.001	-0.002
0.012	-0.008	0.004	-0.063	0.061	-0.002
-0.039	0.004	-0.035	0.023	-0.028	-0.005
0.014	-0.031	-0.017	0.037	-0.059	-0.022
0.018	-0.021	-0.003	0.035	-0.029	0.006
-0.062	0.07	0.008	0.015	-0.031	-0.016
0.006	-0.024	-0.018	0.083	-0.055	0.028
0.064	-0.071	-0.007	-0.068	0.068	0

-0.085	0.051	-0.034	0.041	-0.06	-0.019
0.041	-0.04	0.001	0.014	-0.021	-0.007
-0.02	-0.028	-0.048	-0.08	0.064	-0.016
0.015	-0.028	-0.013	0.031	-0.044	-0.013
0.024	-0.059	-0.035	0.022	-0.03	-0.008
0.045	-0.028	0.017	-0.029	0.03	0.001
-0.067	0.057	-0.01	0.013	-0.024	-0.011
-0.012	-0.003	-0.015	-0.003	-0.025	-0.028
0.018	-0.05	-0.032	0.038	-0.052	-0.014
0.031	-0.015	0.016	-0.026	0.019	-0.007
-0.007	0.004	-0.003	0.037	-0.038	-0.001
-0.001	-0.033	-0.034	-0.025	0.021	-0.004
0.015	-0.01	0.005	0.017	-0.022	-0.005
0.027	-0.035	-0.008	0.021	-0.034	-0.013
-0.038	-0.007	-0.045	-0.002	-0.021	-0.023
0.017	-0.031	-0.014	-0.029	-0.003	-0.032
0.036	-0.055	-0.019	0.054	-0.048	0.006
0.017	-0.032	-0.015	-0.031	0.032	0.001
-0.079	0.098	0.019	0.039	-0.043	-0.004
0.017	-0.02	-0.003	-0.012	-0.021	-0.033
0.093	-0.099	-0.006	-0.007	0.016	0.009
0.047	-0.068	-0.021	-0.006	-0.009	-0.015
-0.075	0.076	0.001	0	0.014	0.014
0.055	-0.041	0.014	0.005	-0.018	-0.013
0.036	-0.023	0.013	0.006	-0.011	-0.005
0.061	-0.057	0.004	-0.048	0.069	0.021
-0.071	0.03	-0.041	-0.039	0.026	-0.013
-0.11	0.02	-0.09	-0.079	0	-0.079