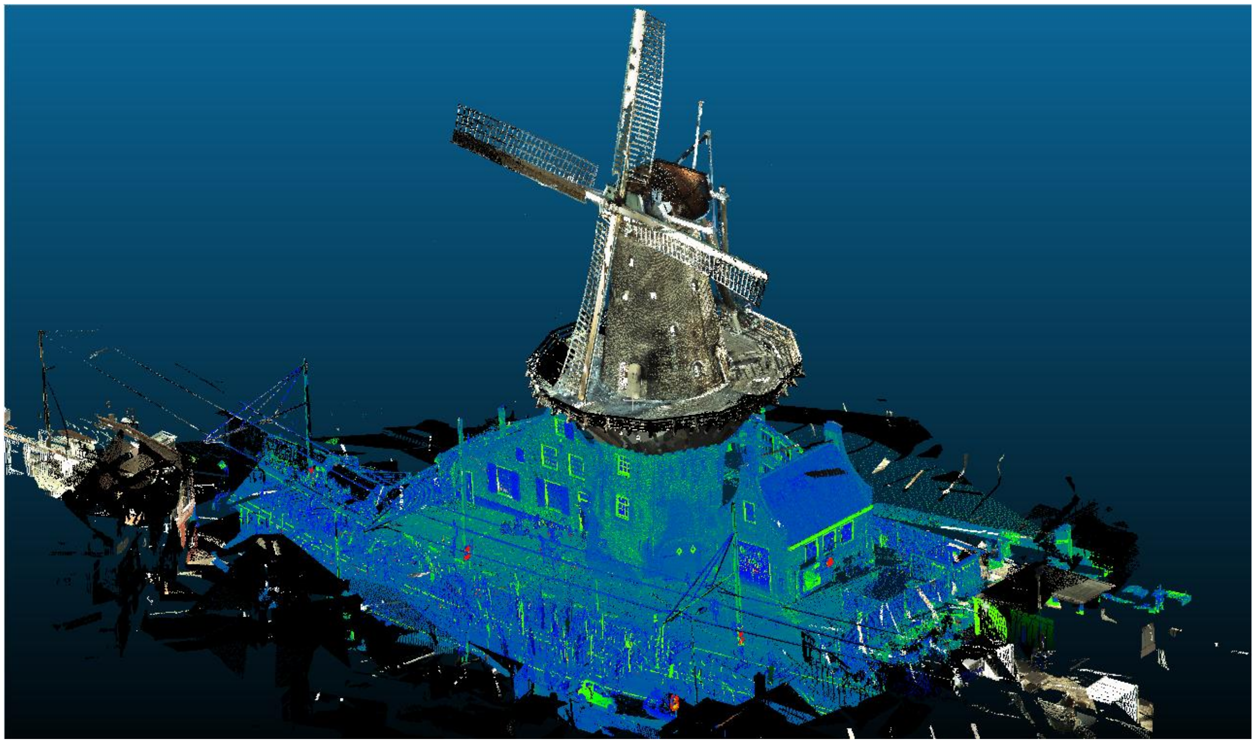
Generating a full 3D Model of The Windmill (Molen de Roos).



Joppe Roebroeks <jopperoebroeks@gmail.com>
Department Geoscience & Remote Sensing
TU Delft, University of Technology
June & July 2015
Supervisors: Dr. Beril Sirmacek
Dr. Martin de Graaf

This report is written by Joppe Roebroeks as being a bachelor Thesis. He is currently a student at the Delft University of Technology. The report and data acquisition have been supervised by Dr. Beril Sirmacek and Dr. Martin de Graaf.

I would like to thank several people, for supporting me and giving me the opportunity to work with the laser scanner. I would like to thank the TU Delft and Dr. Roderik Lindenbergh for giving us the possibility to use to laser scanner for data acquisition and for supporting our research activities even at the equipment transportation stages. I would like to thank Junhu Wang for the exterior point cloud acquisition. I would like to thank BSc. student Adriaan van Natijne for helping with the data acquisition and partial data processing. And at last I would like to thank Dr. Ben Gorte and Dr. Jeroen Lichtenauer for their general support to this bachelor thesis.

Figure on cover page: point cloud model of “Molen de Roos” located in delft represented in 124 million points.

Table of Contents

Abstract

"Molen de Roos" a mill located in Delft, a city in the western part of the Netherlands, is only one of the many old buildings in the Netherlands. More and more these buildings are degrading with respect to integrity, suffering from their age. Therefore it would be desired to look for ways to scan the building in order to find damaged areas. We would be able to repair them before the building is being damaged beyond repair (or very expensively) or before even losing the entire building (through collapse).

In this report there is a focus on documentation of the precise building contour my means of laser scanning and processing the data into a 3D model. Through detailed laser scanning we are able to map a large part of the surface area of the interior and exterior of mill in order to find flaws, direct damage or propagating damage over time.

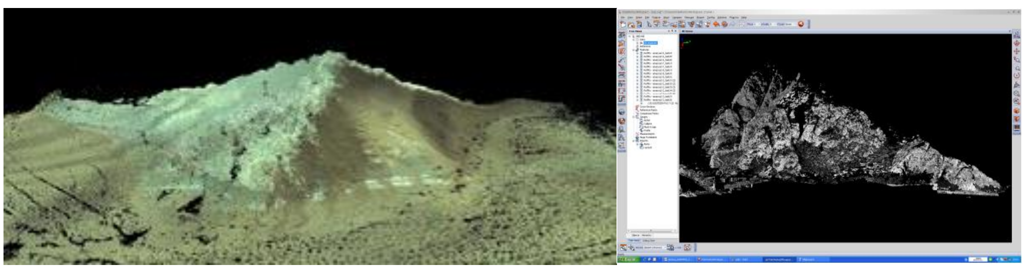
Acquiring data of the full mill takes significant time (there was not enough time to perform a full scan within this project) but gives a very good image of the integrity of the building. In the time available, a believed sufficient level of detail has been obtained. The results of the scanning are promising but are not complete. Such scanning task would take a week if not more. The processing of the exterior scans was rather stressful since the unregistered point clouds had to be registered first. The Interior scans were troubling since the files are very large. With subsampling, reducing the size of individual sub-datasets, the data became easier to manipulate, but at the same time details on the shape of the building was lost. The data also had to be subsampled in order to perform meshing. Meshing was good with respect to floors, walls and roofs, but floating objects in space were rarely recognized by the meshing methods and therefore poorly meshed.

Some problems within this project were of practical nature and some of more programmatic type. A problem is that we are not able to make very detailed point clouds in small amounts of time. Narrow spaces cause many practical problems with respect to scanning areas (resulting in shadows). Computer programs and computers are often not capable of handling the large amount of data produced with such scanners. The algorithms are not fast enough and computers not powerful enough to cope with this amount of data.

1. Introduction

Nowadays a lot of 3D computer modeling is performed; on a large scale and concerning many different subjects, for objects but also how movement of the human body functions. Laser scanning is used to record the current state of an object or area to the finest detail. A car can be scanned with 10’s of millions of points easily. Below some examples are shown where laser scanning has been used.

a b c



c c

Figure 1: different applications laser scanning, ref: (1),(2),(2),(3),(4)

Laser scanning is used in many applications. Above examples are shown for a car and a railroad tunnel in figure 1a, b and c. Laser scanning is very useful in these applications because one can easily find imperfections. Scanning is often used to find changes of shape over time or as a method to improve an object (car or railroad in this case).

Also the change of shape of dunes or mountains (see figure 1c and d) over time can be studied with laser scanning and subsequent data / file comparison. When using geo-referencing you can look what the changes are when scanning the same area, over a longer period of time. This can be extremely useful when looking at structures. Questions answered with this technique can be if the structures are safe, are they to be endangered in the future, are they eroding, or if large plateau’s slowly sliding down a hill. All of these features can be captured by laser scanning, if done correct.

This report is focusing on the modeling of a windmill. Many different objects have been modeled, but windmills have never. Especially this windmill has experienced quite much since it was first build in 1679 (5). The windmill was initially built on soft soils so it could somewhat move when there were stresses on the mill. But later in its life, more traffic came, with a tramline and a railroad next to it. Very recent the railroad even went underground. Therefore the windmill had to be lifted from its foundations and was relocated. Today the windmill is standing on the railroad tunnel (bound to it). All these physical challenges and changes loaded the windmill structure to its limits and beyond. Cracks occur and solutions have to be found since changes on old buildings are being performed more and more. Much of our old culture will be affected by the future, and therefore we need to start thinking about conservation of our cultural heritage. Our buildings are important to our history, culture and tourism. Moreover most of the mills still have a function, and therefore it is especially important that we start thinking on saving our old buildings instead of just building the new ones.

Laser scanning and modeling will become more important since they can provide a lot of information on the condition of our old buildings. This report is new on this ground. Actual laser scanning of windmills have been performed a few times, but creating a full 3D model together with removing outliers complete in color, is something that has probably not been done before or has not been published (no reports on this subject have been found).

This report will focus on making a full 3D model of the windmill “Molen de Roos” located in Delft. A full 3D model means that the exterior scans and interior scans are merged to gain the most complete model possible. This model is made out of two datasets of scans; one dataset is made by a PhD student at the University of Delft, which will be used in this report as the exterior of the windmill. The interior scan is added here in order to make a full model of the windmill. The scan of the exterior of the building has a lower point density than that of the interior. The difference in point density will be visible in the model. The result will be a fully computed 3D mesh model and slightly smoothened. Finally a video is created allowing a walk through the model.

The difficulty in the project is that we’re dealing with an object which is not rectangular; it has very few squared objects since almost every object is designed for turning. The turning parts cause a problem because not all of the scans are made on the same day. Therefore the top of the mill can change in position, causing problems in registering the point clouds. Also scanning inside a mill is a challenge on itself. The rooms are pretty small and narrow and covering of all the rooms with scans is hard if not impossible because of the large amount of objects in a small room (keeping in mind that there is only a limited amount of time). Linking the floors is a problem too since the stairs are very steep, the openings are very narrow and there are mostly only 2 openings are available per floor. The device has a maximum scanning dip of 45 degrees down (it cannot scan under a greater angle than 45 degrees down from the horizon). So linking the floors of a mill was a very big practical problem.

2. Method

When performing a project using models in programs, it is important to keep track of every step and to gather information about previous investigations about this subject. There is hardly any information about scanning Mills (electricity producing mills or Windmills more specific). The only data on previous activities is scanning of the inside of a mill in England in Cambridgeshire. The site of the mill scan can be found under reference (6). There might also have been scans of electric mills, but they have most likely not been reported.

The only available data is from the exterior scans from the windmill here in delft. These scans have been performed by Jinhu Wang (7). This dataset consists of 14 scans of the exterior of the windmill from different angles all around the building

In the following paragraph the methods which are used to perform all the different operations within Cloudcompare and Meshlab can be found. It describes how the buttons of the program work. Finally a workflow sheet is provided of all the actions performed when processing was done.

2.1 Specification of the scanner.

For the Interior data, first the data needs to be acquired. This is done with the Leica c10 Scanstation. The official specifications of this laser scanning station mention that it is a compact, pulsed, dual-axis compensated, very high speed laser scanner. It has a survey accuracy range and field-of-view with integrated camera and laser plummet. The system scans every single measurement precise to 6mm with respect to position and 4mm precise with respect to range. The device has a standard deviation of 2mm. This precision is probably defined for a certain range, but this range is not defined. Probably the scanner is much more precise when scanning at close range (in the order of a distance up to 5meter). The maximum scanning distance or range is 300 meters and the minimum range is 0.1 meter. The scanner scans a maximum of 50.000 points per second at maximum instantaneous rate. The laser is a 532nm green visible laser. There is a 4Megapixel camera with a resolution of 1920x1920 which captures colors in order to give color to the laser scan. All the data is stored on an 80GB solid state drive (SSD). For full specifications of the device I would advise to look at reference (13)

2.2 Cropping point cloud for selecting the object of interest.

The raw data includes a significant amount of data which is not needed for processing and will severely slow down the processing. Therefore the data needs to be cut so that only the mill remains (with some reference in order to be able to align the clouds). The principle is as follows. In the program you can select a 2D plane area where you want the information to be kept or removed. But in 3D the area extents into the screen infinitely, it is shown below in a picture below, made with paint.

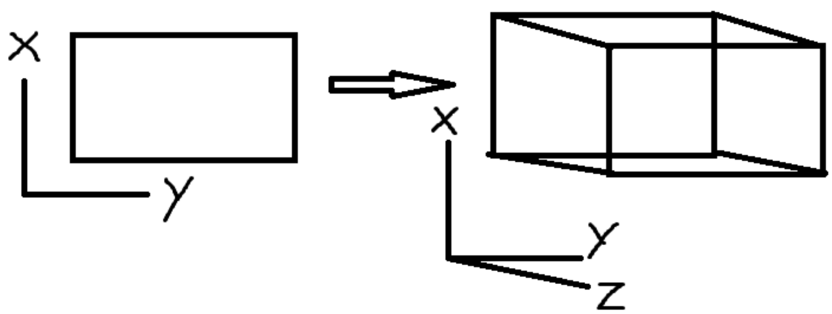


Figure 2 visualization cutting box

An area in the x-y plane is selected, but actually a volume in the x-y-z space is chosen, which extents infinitely in the z direction. After choosing the area the points inside can be chosen to remain, or the points outside the box can be chosen to remain. With this option in Cloudcompare, data is removed that is to be excluded in the analysis.

2.3 Aligning.

It is important to make sure that the exterior data scans are aligned, contributing to the same object. This aligning has been done using Cloudcompare v2.6.1 using the manual aligning option using equivalent point pairs. There are actually three different registering methods: Matching bounding box centers, Point pairs picking and ICP. Matching bounding box centers is a method which does not work on this point cloud since almost every cloud is rotated while this method cannot perform a rotation. It would save some clicks but the method will not align the point clouds completely. The ICP or Iterative Closets Point method works with two point clouds which have to be aligned. The pre-alignment needs to be very good in order to work with the ICP. Our best option is the point picking method. With this option common points in separated clouds are chosen. When (in theory) 3 or more of these common points have been found (the program wants at least 4) a point in space has been defined. Generally the clouds used in this report, are aligned with 10 common pairs.

Of course when choosing common points, the same points in different clouds cannot be selected twice. This is nearly impossible since the point density is different and the same point is rarely sampled twice. This causes a problem in the aligning because different points of the different clouds will never fit exactly on top of each other. The computer will try to fit the clouds on top of each other with the smallest error called the RMSE or Root Mean Squared Error given by the formula below. This formula is used in Cloudcompare and referenced under ref. (8) for further information.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Equation 1

In which:
n=number of points chosen
y_i-$\hat{y}_i$= distance between the two points

In this formula the largest difference in the points will cause a large error. Therefore the program tries to minimize the RMSE causing the chosen common point pairs with optimal fit.

2.4 Subsampling.

Targets for the laser scanner to be recognized, have been allocated all around the building in order to make it easier to link the different scans later on in the process. The scans will be linked through 3 (or more) targets in every scan, of which at least 3 are also visible in other scans. Later these targets are used to give them a specific point in space. When all the scans are made, they can be processed by Cyclone in order to link the scans and combine them in one big file. Thus they are orientated in a right way. Then they can be put into Cloudcompare in order to downgrade the point density, reducing the size of the file. The subsampling option within Cloudcompare is used for this. Initially a subsampling was used where the minimal distance between two adjacent points was 0.002 meters. This means a maximum of 1 point every 2mm along a line and a maximum of 25 points per square centimeter (calculation: (1/0.2)^2=25). In Cloudcompare the option is described for sampling of points where the minimal distance between two points is equal to an entered value. So entering a 0.002 means the points in the cloud will be at least 2mm away from each other. This was used because while scanning the same floor a few times, it appeared highly likely that the same point is scanned twice. Because of this double sampling, too much detail of a specific area is acquired and loading the extra points costs a quite some additional memory and may not be doable for most computers.

2.5 Outlier removal for noise reduction.

The next step is the removal of noise in the clouds also using Cloudcompare with the statistical outlier remover. The removing of the outliers is a very accurate business. It is not even sure that outlier removal will give results which are desirable for the model. When the statistical outlier remover in Cloudcompare is chosen, for every point in the cloud the mean distance to the x nearest points is determined. The amount of points x points are to be defined by hand, and the mean is the average of the largest and smallest distance to the point to be calculated. Also the standard deviation of those x nearest points will be calculated through the formula below. In reference (9) further information about the outlier removal technique used at Cloudcompare can be found.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Equation 2

In which:

σ=standard deviation.

N=number of points to be compared with (for example nearest 1000 points).

x_i=distance to point i.

μ=average distance of all points to the point to be compared with.

After the mean and the standard deviation are found, another value “y” is filled in defining the cutoff for the standard deviation. The program will then exclude the point for which you’ve calculated the standard deviation and the mean, if at a distance further than y times the standard deviation of the mean distance. After defining different values for the x and y, the best possible solution can be observed.

2.6 Color interpolation.

Also on the Interior scans (and partially exterior) color interpolation has been performed in order to give color to the data points which only had scalar colors so far. Scalar colors are not the real colors. This interpolation has been done with Cloudcompare with the “interpolate from another entity” option. This interpolation is actually a very simple algorithm; it looks for the nearest point and takes that color in order to color the uncolored point. It is called the “nearest point” method, and it uses vonoroi diagrams in order to determine the nearest color point for the uncolored data point.

2.7 Generating 3D surfaces.

2.7.1. Delaunay triangulation

It is investigated if some mesh generation algorithms can be used in order to gain results regarding triangulations. First the Delaunay triangulation is used. Delaunay triangulations are based on the fact that the algorithm tried to find three points (and makes a triangle) in which the angle in every corner within the triangle, is as large as possible.

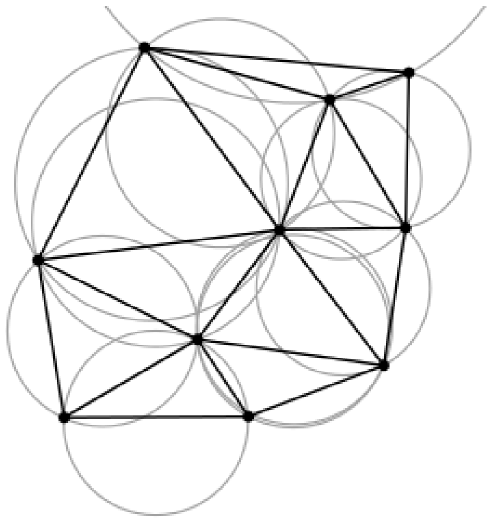


Figure 3 example triangulation steps including circumcircles [11]

The image above shows how the Delaunay triangulation works. The algorithm fits a circle through 3 points, as large as possible, without having any other point with the circle, the points on the circle will then have the largest possible individual angles. This method avoids skinny triangles. For further references on the Delaunay triangulation look at (11).

2.7.2 Ball pivoting mesh generation.

If this triangulation does not work properly which is likely (no triangulation is perfect), meshing in Meshlab using the ball-pivoting method could be tried. This method is relatively simple, effective and powerful. Assume a 3D object M with sampling S of the surface of the object. Now 3 points are chosen fitting a circle with a certain radius p through those three points (the first triangle). Then pivoting the around two (of the three) points from the beginning is performed. The next point (3rd point) encountered will then belong to the first two points chosen to pivot around, forming the second triangle. The visualization of this process in 2D is shown in figure 4 below.

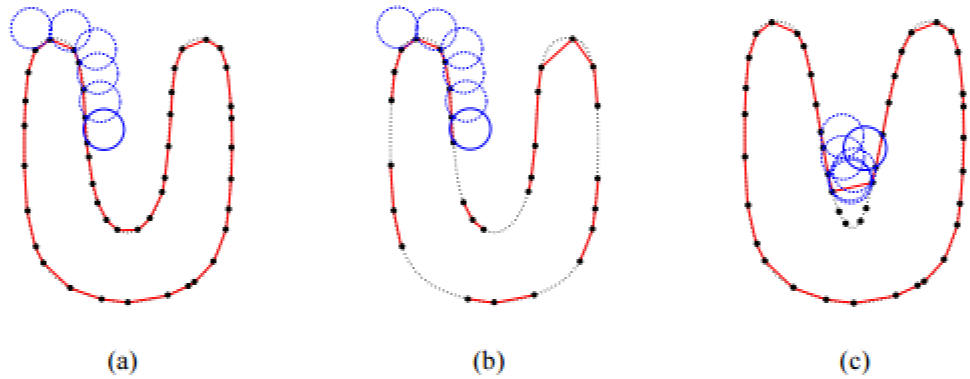
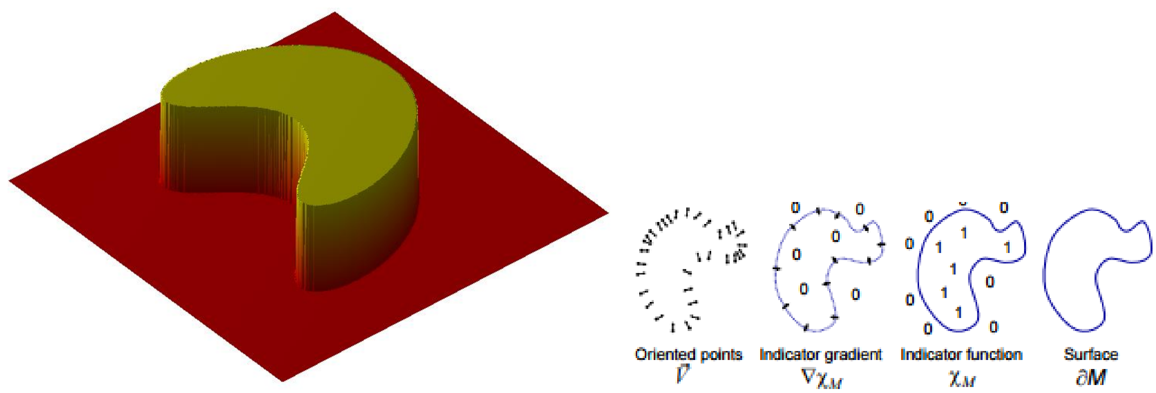


Figure 4: ball pivoting principle in 2D

In the figure above (referenced in reference 12) is shown that evenly sampled objects are easily processed by the ball-pivoting process. Unevenly sampled objects are very hard to handle and will give holes in the mesh. For curvatures which are tighter than 1/p (half the radius of the ball) problems will occur creating holes in the mesh. This ball pivoting method is implemented In Meshlab, but in Meshlab it is not even near as fast as suggested in the article about ball pivoting and computing of the mesh (12).

2.7.3 Poisson reconstruction mesh generation

The Poisson reconstruction approach is an approach based on the relation between the orientation of points in space and the indication function of a model. An indication model defines which space or void or actually a set of points sampled, belongs to an element or subset. An example is shown below.



a b

Figure 5a and 5b: example indication function resp. visualization of the Poisson approach

The value of the indication function is zero almost everywhere except at the surface of an object or points of an object, where it is equal to the inward normal of the surface at that point. Now the points at the surface can be viewed as sampled of the gradient of the model’s indicator function. Now this gradient needs to be inverted so that eventually a potential surface area is being created by using the standard Poisson problem: computing a scalar function whose Laplacian equals the divergence of the vector field. For further information about the Poisson approach, look at reference (18)

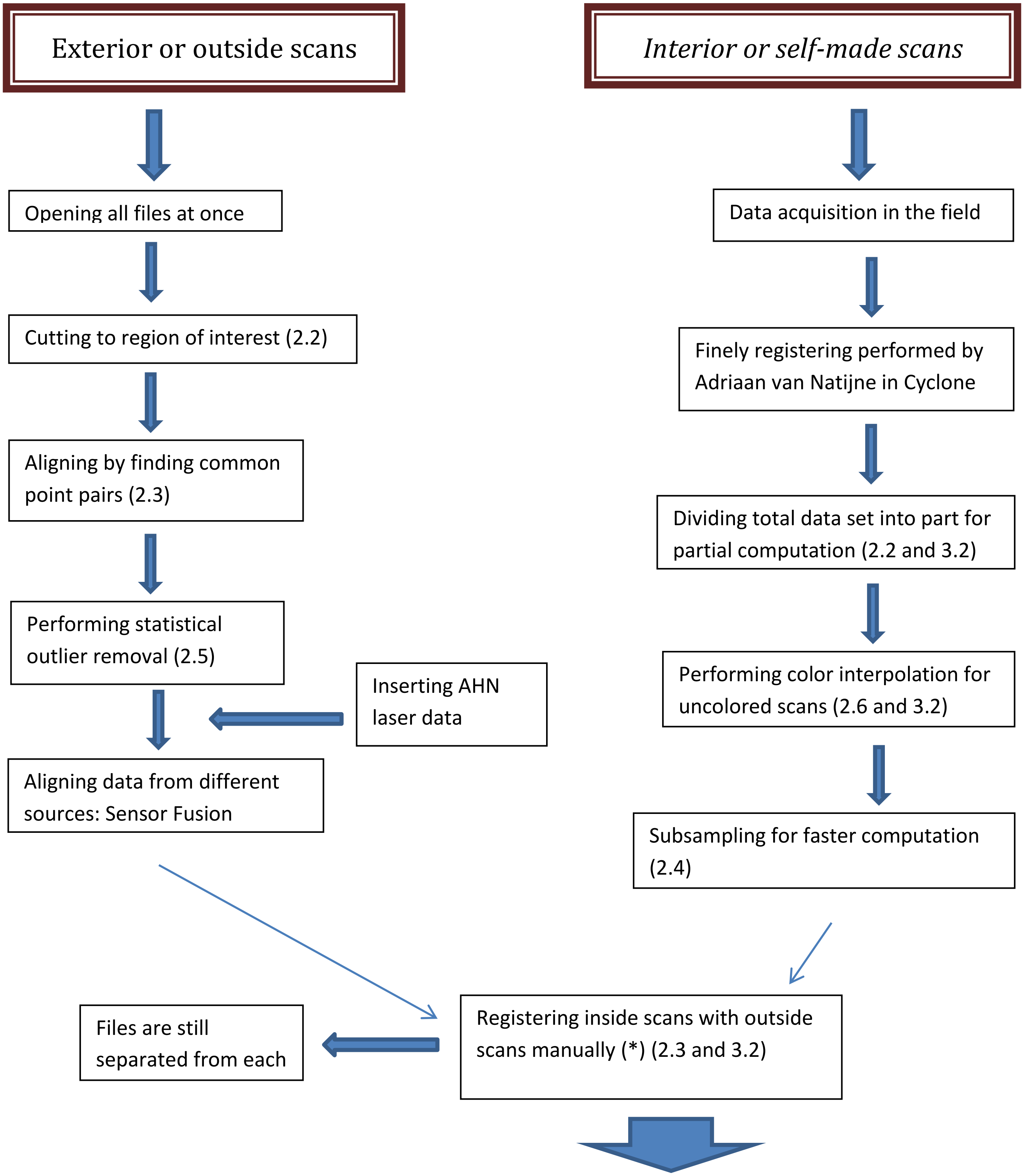
2.8 Flow Chart

On the next page, a flow chart is provided. This flow chart is an easy guide through the different steps taken in the processing. It starts with two separate data files called the exterior scan and interior scan. The exterior scan is made by someone else and is not aligned.

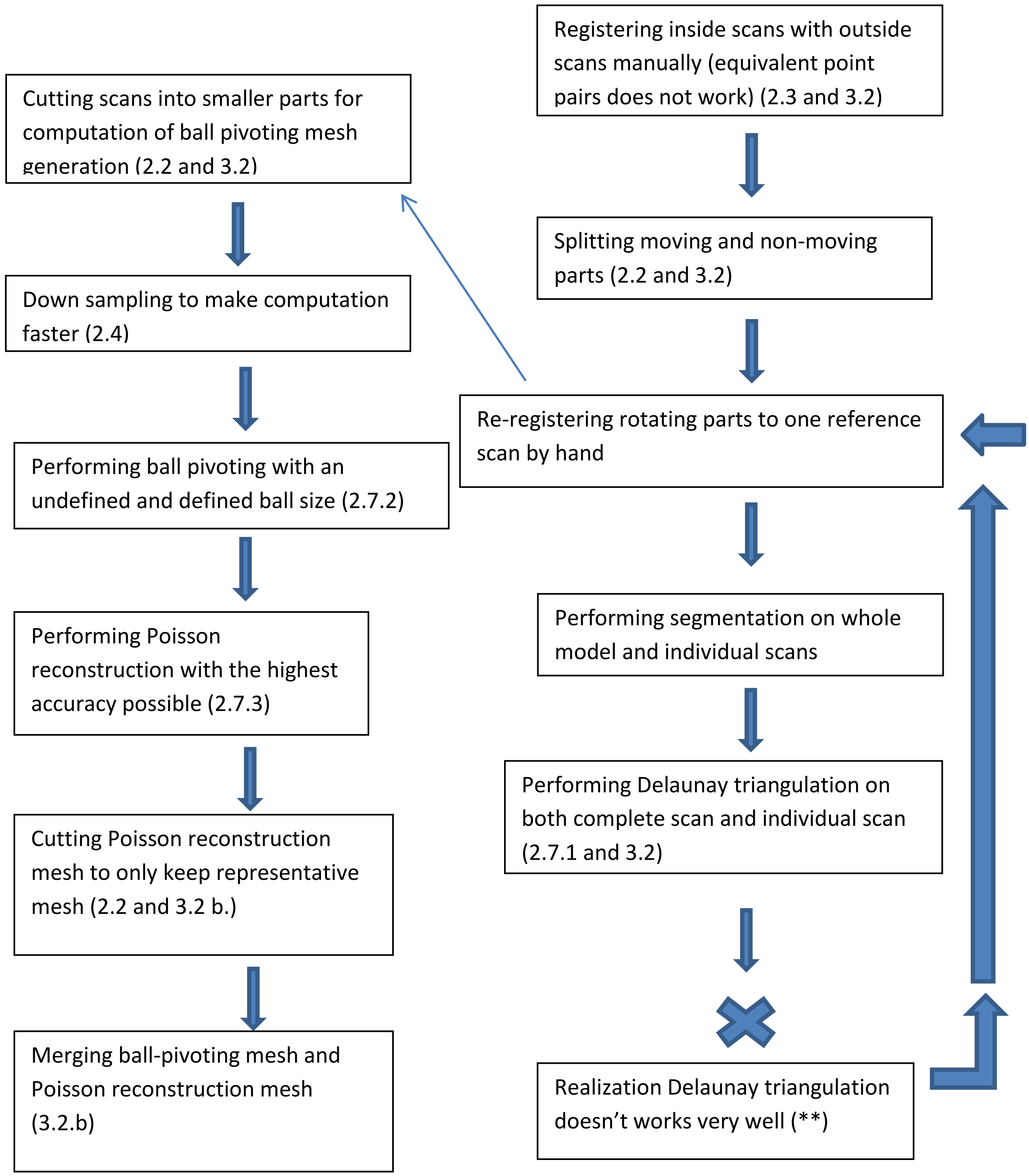
First the exterior data is cut. Then the cut data is aligned through picking common point pairs. Then statistical outlier removal has been performed to look if noise could be removed. Finally the exterior scan was merged with AHN data because the roof was missing; this action is called Sensor Fusion.

The interior data had to be acquired first. Then registering has been performed by Adriaan van Natijne, making sure the scans were aligned properly. Then the scans were divided in parts (floors and balcony) so that partial computation could be performed. Color interpolation was performed for clouds which had no color. After every section was fully colored, subsampling was performed to reduce the amount of memory the clouds would take.

Then the exterior scans and interior scans are aligned manually in order to create one coordinate system which represents both the inside and the outside of the building. The fully aligned point cloud was then cut again in order to separate moving and non-moving parts. These moving parts were then re-aligned in order to meet one reference. Segmentation was performed together with Delaunay triangulation, on the individual scans to see what the result would become. After that even more down sampling was performed and more cutting was done in order to import the data into Meshlab. In Meshlab Ball pivoting and Poisson reconstruction has been performed. The Poisson reconstruction mesh had to be cut again since the full mesh was not fully representative for the object.


Exterior or outside scans
Opening all files at once
Cutting to region of interest (2.2)
Aligning by finding common point pairs (2.3)
Performing statistical outlier removal (2.5)
Inserting AHN laser data
Aligning data from different sources: Sensor Fusion
Interior or self-made scans
Data acquisition in the field
Finely registering performed by Adriaan van Natijne in Cyclone
Dividing total data set into part for partial computation (2.2 and 3.2)
Performing color interpolation for uncolored scans (2.6 and 3.2)
Subsampling for faster computation (2.4)
Registering inside scans with outside scans manually (*) (2.3 and 3.2)
Files are still separated from each

Flow Chart



(*) The point pair picking tool does not work, probably because the point density is too great for the tool.
(**) the results of this triangulation are shown in the results regarding Delaunay triangulation under chapter 3.2. a.

3. Results

3.1 Exterior scans.

This part of the report will describe results gained from the laser scanning and processing of the laser scans. First of all, the exterior scans, acquired by someone else, will be evaluated and processed. Then the Interior scans which are made by the author of this report will be discussed.

When opening the exterior scans (there are 14 of them) it is observed immediately that objects are positioned in a strange way; not aligned as shown below.

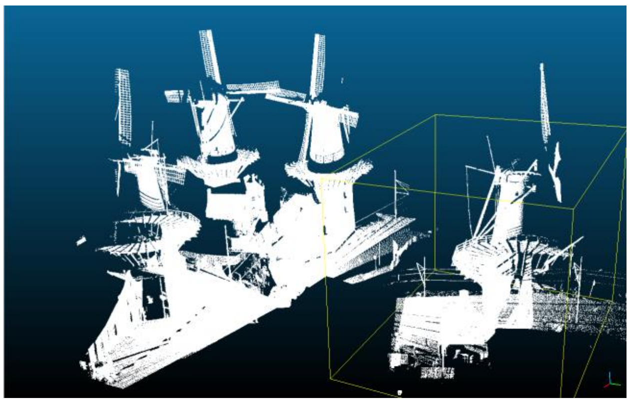


Figure 6: data is not aligned, 11.7 million points.

So the first step would be to merge the clouds to make 1 big cloud, representing the whole windmill. When trying to merge all clouds to one (there were 14 clouds in total) the problem occurred that the point clouds all represented a different angle of the windmill. They are all made with a certain point density, sometimes a low density and sometimes a high density.

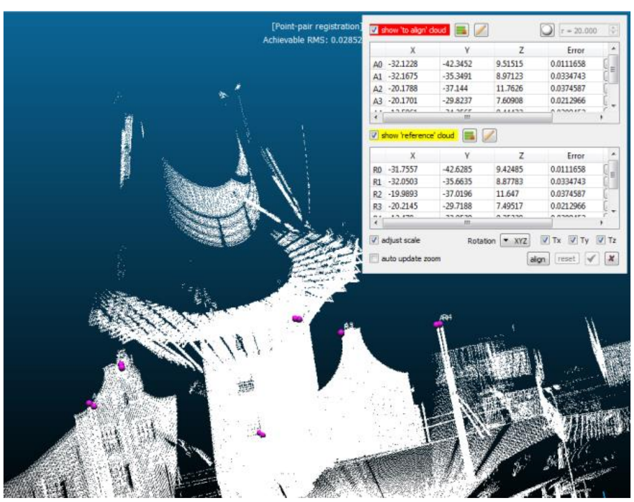


Figure 7: aligning through equivalent point pairs, varying 300 thousand (300k) and 1million points.

It was a true challenge to find common points pairs in some clouds. Sometimes points had to be taken which were most likely to be common. The program will automatically make the best possible fitting overlay of the chosen common pairs, with the smallest error. The program chooses the best overlay, and that causes for some irregularities in the end model as indicated in figure 8.

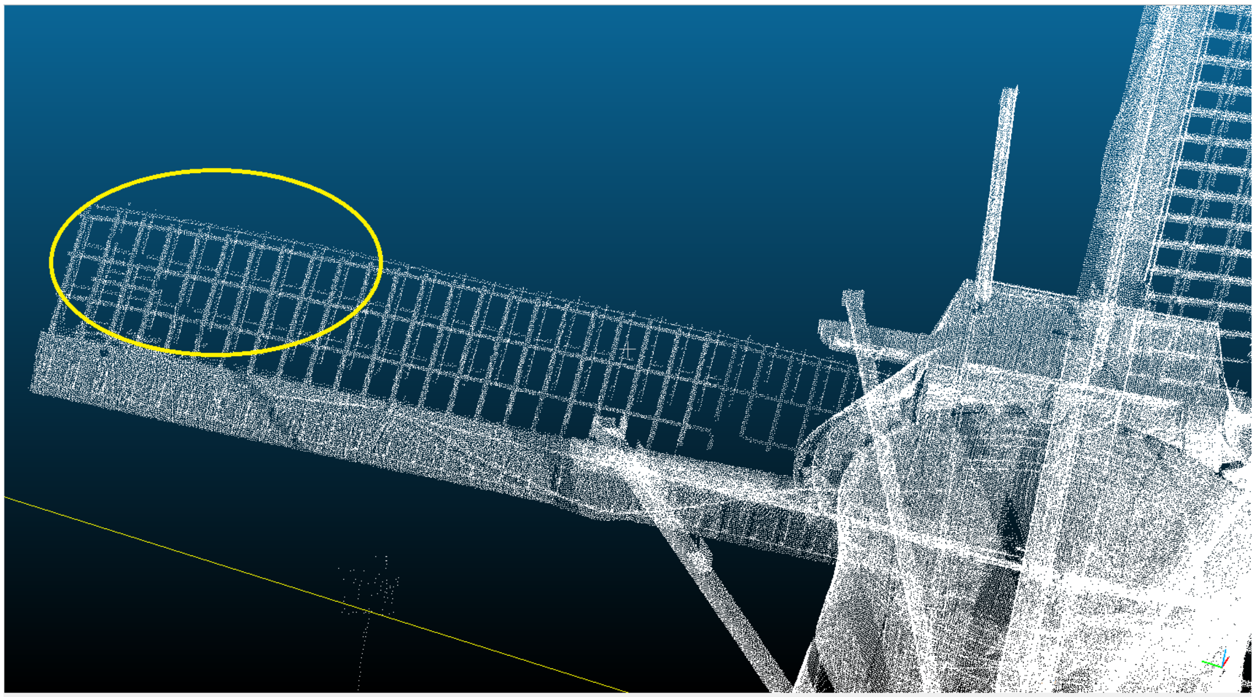
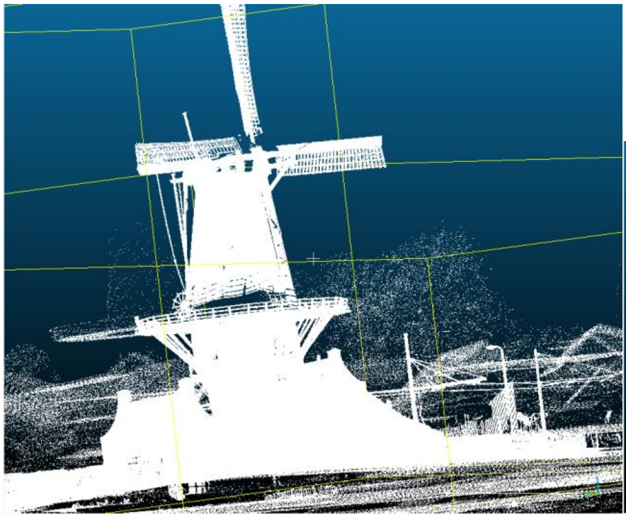


Figure 8: error in alignment, total model 11.7 million points.

This figure shows that there is some strange overlay in the blades, sometimes that is something we'll have to account for. The exterior scan is in general less good than the Interior scan; less accurate and with manual merging of the clouds. Also the roof is missing in the scan, because the scan is made from below. Scanning from below gives indeed the implication that the topside of a building can't be captured. Governmental airplanes have performed some laser scanning as well. These governmental scans are laser scans belonging to AHN. AHN or "Algemene Hoogtebestand Nederland" is a digitalized height model of the Netherlands. The data used comes from the project AHN2 which started in 2007. This data is a point cloud with a density off 10 points per square meter (pts/m^2). Those laser scans have been used to fill up the gap on the roof of the windmill as good as possible. But the outdoor scan was not geo-referenced while the airplane scan was geo-referenced. This means there is a shift in those two laser scans as well shown in the picture below. Because of this shift the so called Sensor Fusion has to be performed: merging data from different sources in order to make a complete model of the same object. The governmental scan of the vicinity of the mill is around 16000 points rich; this is considerably less than the self-made scan of 11 million points. The data fusion was therefore also not too easy and had to be done carefully by hand.



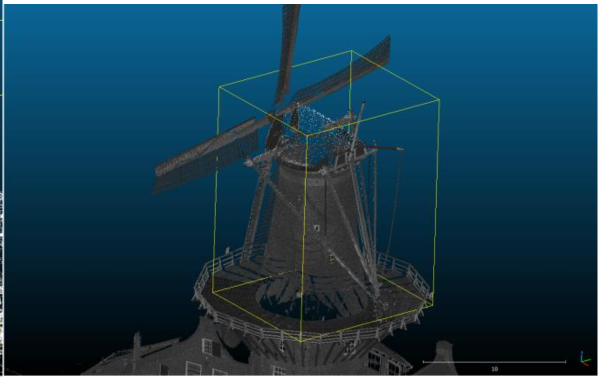


Figure 9 & 10: different point densities visible

The final result of the merging process shows that different areas of the mill are represented with different point densities or at some locations the points are totally missing (this because the scanner was not able to "see" these positions).

Also the clouds could not be merged perfectly, so eventually outlier removal is performed in order to look if some data can be filtered out. Different scenarios have been evaluated, and all these scenarios had a problem. The roof of the mill could not be scanned, so airplane information has been used to fill up the roof with some data points. The density of the roof is considerably lower than the density of the rest of the mill. The roof of the mill has therefore been excluded from the outlier removal by adding that part of the mill after outlier removal has been done.

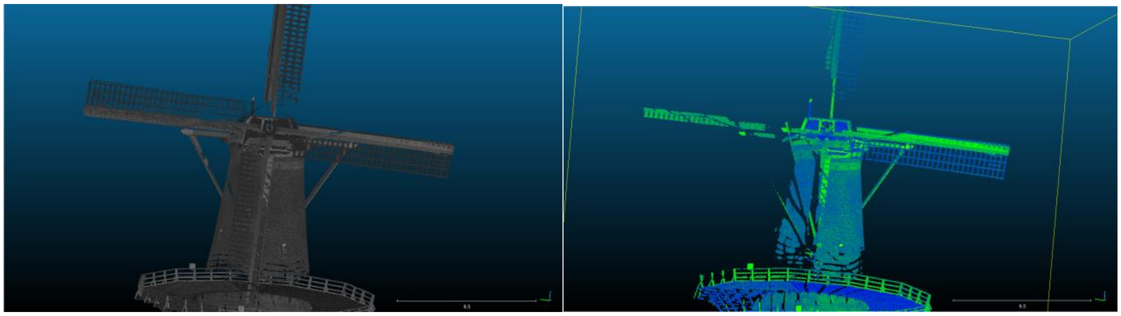


Figure 11 & 12: original cloud & SOR with 10 points and standard deviation 1, resp. 11mil points and 9 million points

A lot of different scenarios have been performed in order to find the ideal solution for the outlier removal, concerning the amount of points used for determining standard deviation and the value used for the cutoff value (mean +/- c*standard deviation with c being a constant value that needs to be determined). The determination of this constant value c is more or less arbitrary. The values for c are derived from statistics using a normal distribution. The standard deviation is determined assuming the distance to one specific point is normally distributed. On the internet information on normal distributions can be found. But in the statistical outlier removal, standard values of 1, 1.5, 2, 2.5 and 3 have been used for c in order to consider the influence of the standard deviation. What is found is that as soon as the program starts to remove outliers, it always removes crucial points. Therefore it is a better idea to not remove outliers in the data, and just to account for them when creating a mesh model. Applying smoothing algorithms might take care of the fact that outliers are present in the data.

3.2 Interior scans.

Data acquisition for the data processing was the first step It has been decided to use targets inside so that the link between the scans would be as accurate as possible. Different target sizes were used in the process, and almost all of the targets were recognized by the scanner itself (the absolute center of the target) so that linking between the targets was as easy as possible. Sometimes the target centers had to be determined by hand because the angle under which the center was determined was too small (the laser beam was too parallel to the paper target surface). In order to define a location in space, it had to be sure that there were at least 3 targets visible on each scan so it would be clear where these were in space.

From each floor to the other, at least 3 targets were needed as well, visible from at least one floor. Because there were quite some beams in the top of the building, shadows easily occurred where scanning was impossible. In the first day it took about 6 hours' time, to scan only the top two floors, because the positioning of the targets and scanner were much of a hassle. Also marking the exact position of the last scan needed extra attention to prevent a problem the next time you start scanning again.

The alignment and putting it into a .ptx file has been done by Adriaan van Natijne since he has a lot of experience in cyclone and knows how to extract the data from the scanner itself. He provided the total scan merged together.

So the scans of the exterior are about 140 megabytes in size for all of them together. The Interior scans were about 700 megabytes individually. This is a significant file size to handle for the computer; it has been scaled down in order to make it easier for the computer to compute. Also for the project scope the very high density is not necessary since we are not analyzing fine building details such as the cracks in the wood. A full model will be made in order to be as complete as possible, but for a first computation the density has been scaled down to a maximum of 25 points per square centimeter; sufficient density to work with. The areas very close to the position of the scanner itself where affected the most since not everywhere, the density was greater than 25points per square centimeter. The difference is shown below.

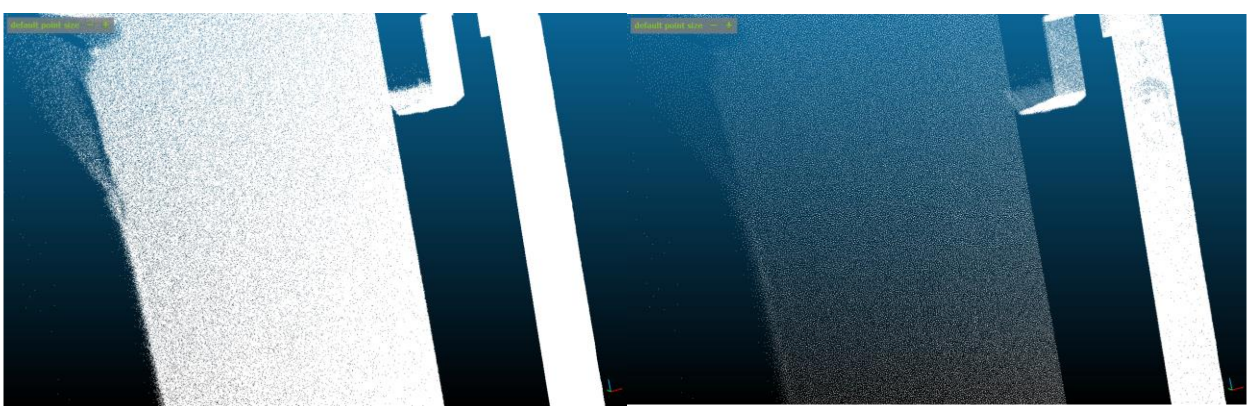


Figure 13 & 14: difference shown original point density and new point density of 5 points/cm

Most of the scans were made with color, meaning that the color could be toggled on and off in the scan. As said, most of the scans have color, but not all. The scans which had no color have been colored by interpolating the color from a different scan nearby.

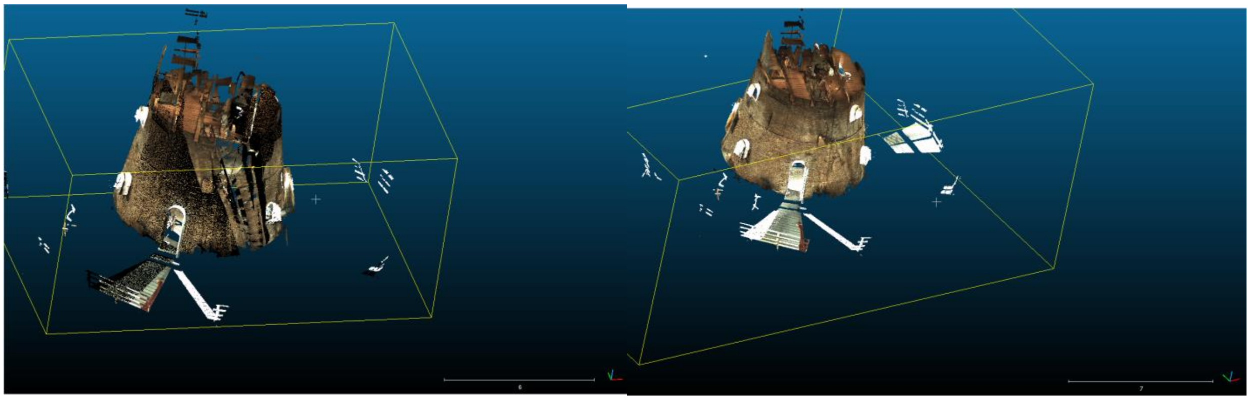


Figure 15 & 16: coloring problems concerning uncolored clouds (both 58 million points)

Like this the desired scan will get a color. This coloring takes a lot of computing time from the computer. Changing to colors from scalar to real colors actually gives the color difference shown in the figures below.

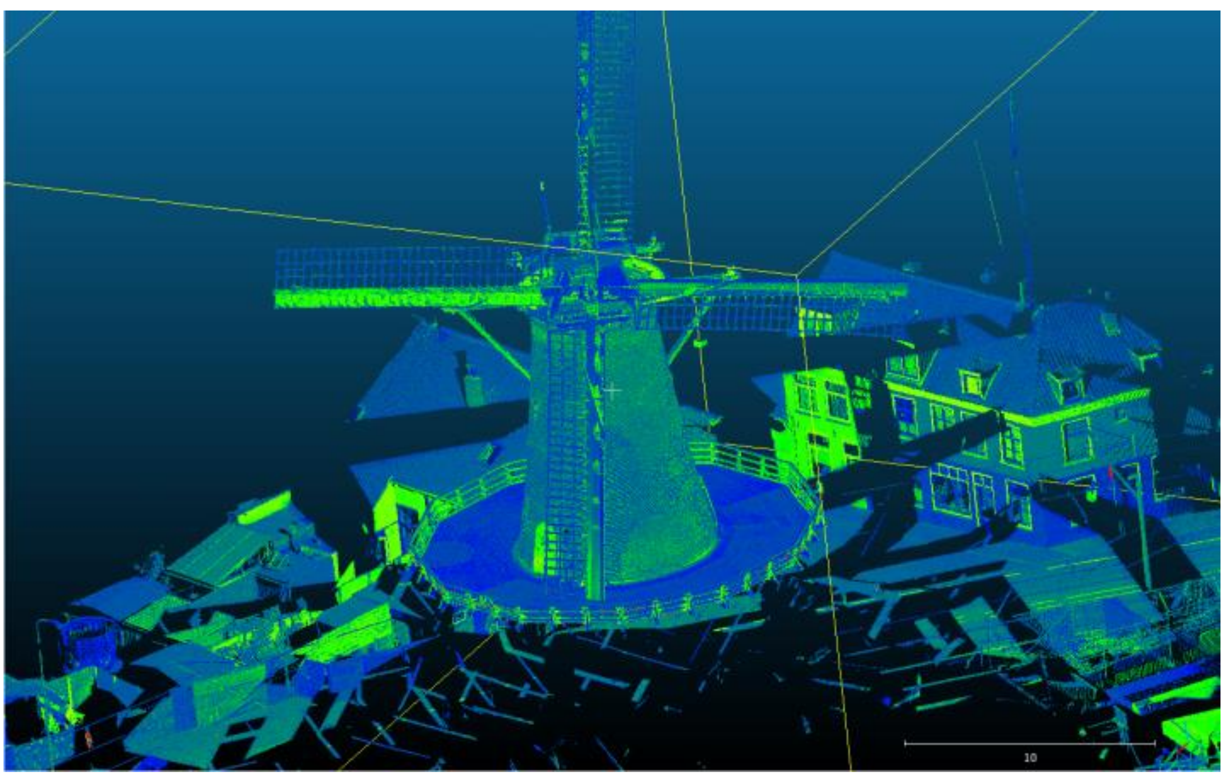


Figure 17: scalar field picture.

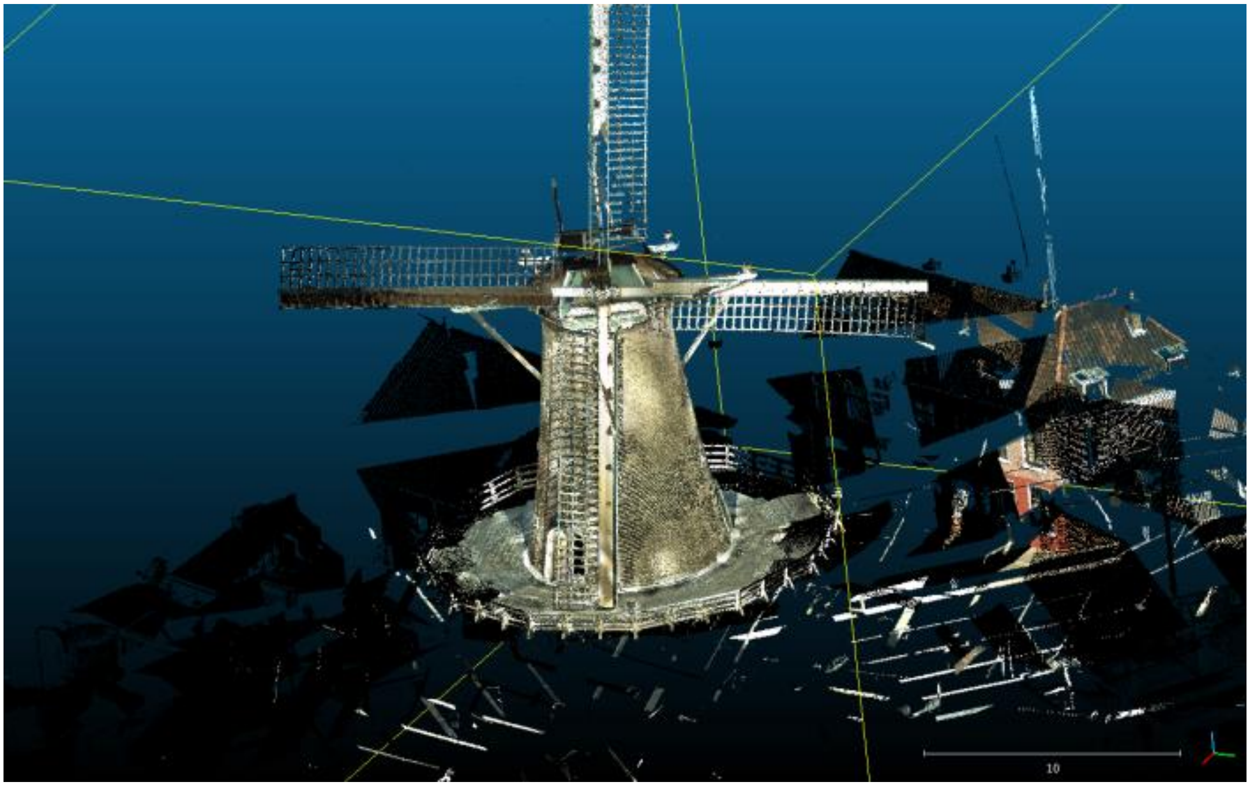


Figure 18: color field picture.

Figures 17 and 18 show a significant amount of data generated around the windmill which might be interesting for the analysis. However such analysis is out of the scope of this project. Also the scans made on the balcony were giving some results. Figure 19 bellow shows that a data line is created, in a straight line above the scanner. This data line is a result of the carrying handle of the scanner which has not been removed. The final model must not include these strange ornaments, so it has to be removed as well.

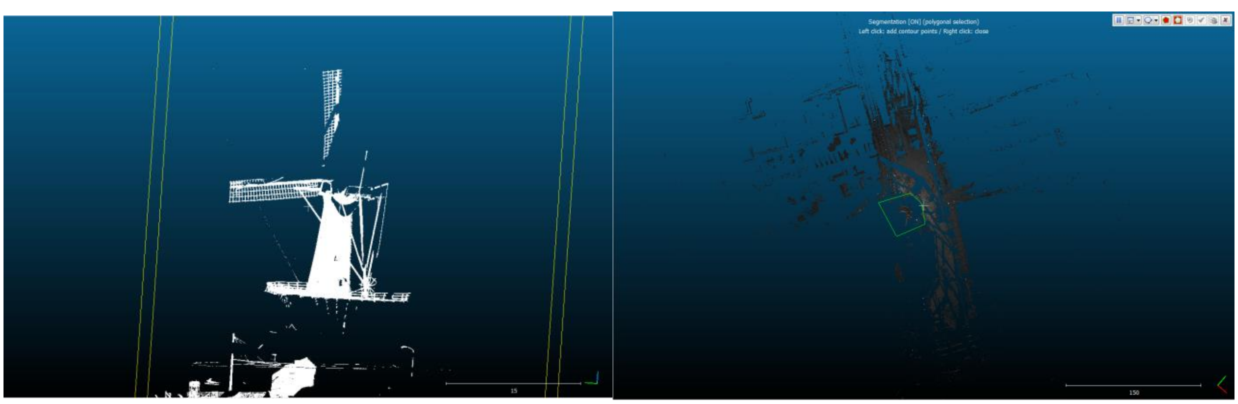


Figure 19 & 20: removing ornaments and cutting balcony scans, 4.9million points resp. 6.4million points

The removing of the ornaments and the removal of the environment around the windmill eventually provided the following result of the cut balcony in color.

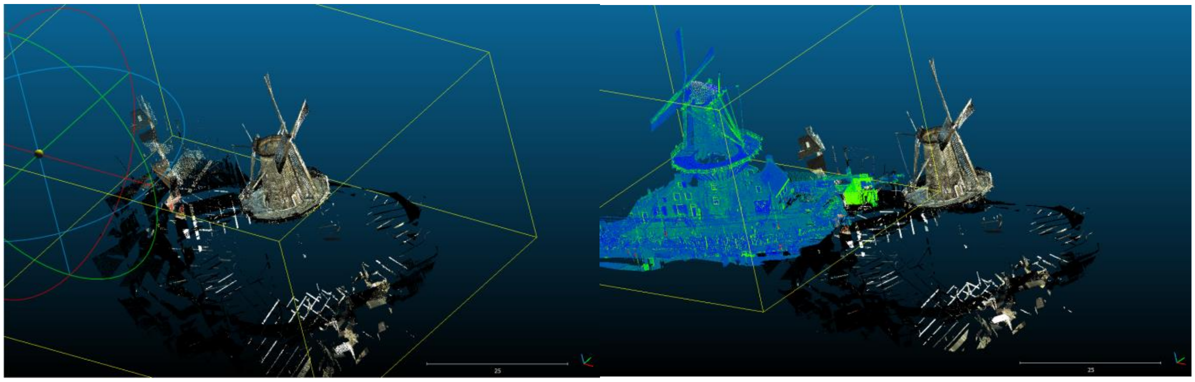


Figure 21 & 22: appearent result and reality (114 resp. 124 million points)

Now that the separate file blocks have been cut and properly colored, it is time to try and open the different files all together and see if they are aligned properly, from the picture above right is observed that the clouds are not even close. The clouds are turned and relocated. The scans are made at different moments in time. This causes the blades to be turned in a different position as shown in the right part of figure 21&22. Only a part of the exterior can be merged since the blades are in a different position. What needs to be done is cutting the turning parts from the exterior and transform that part in order to fit with the inside of the mill since that is taken as reference. The result is shown on the following pages, where the turning parts of the exteriors are detached from the static part of the body. This detaching is done by several cutting procedures since it is not possible to cut the turning part at once.

On this page the results of the different cuttings and the final results of the colored point clouds will be shown.

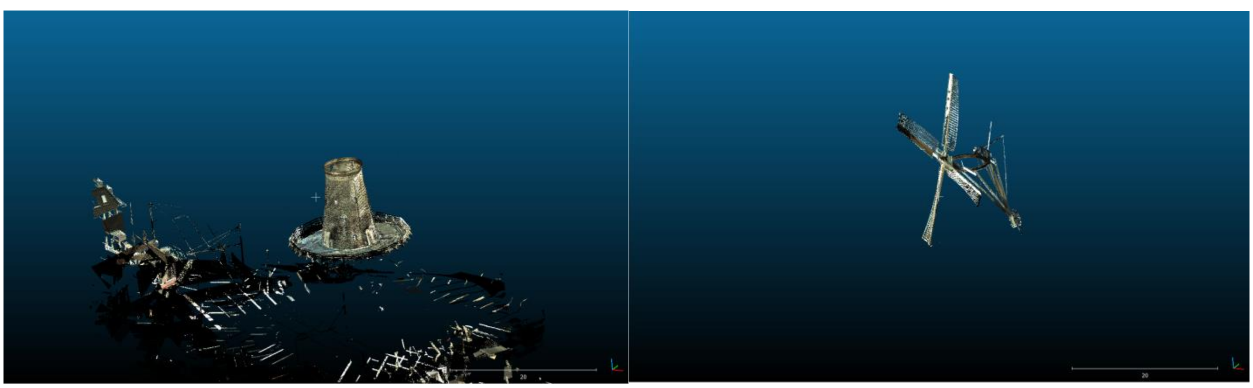


Figure 23 & 24: cutting balcony in parts (resp. 36 million and about 4.5 million points)

Figures 23 & 24 show the cuts from the balcony scans, where the turning part has been detached from the static part of the mill.

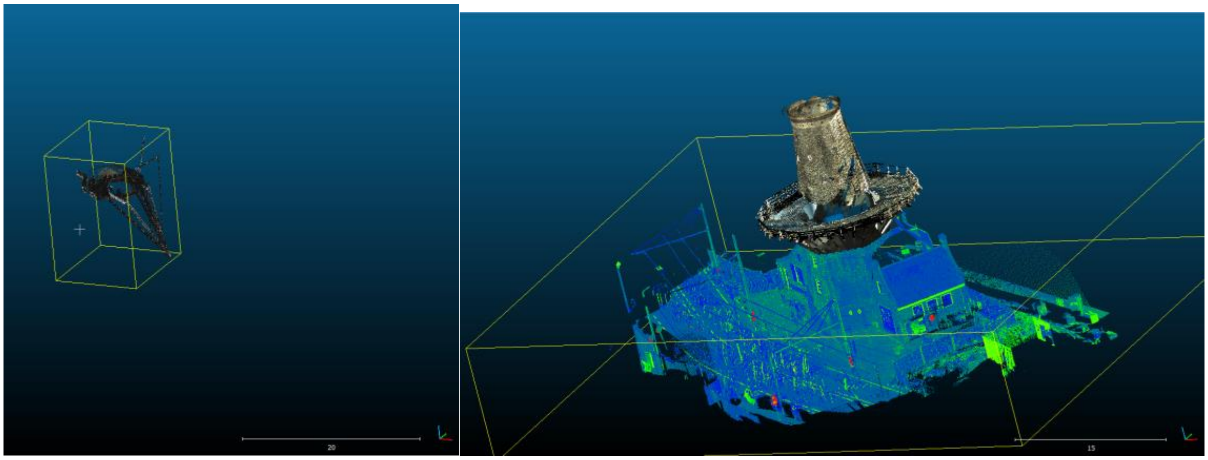


Figure 25 & 26: exterior scans cut and remaining part exterior scans

Figures 25 & 26 provide the governmental areal scans combined with the previous scans done by another student, also cut. The topside of the mill has been colored together with the balcony. This color has been interpolated from the balcony scans since those scans have color and these scans are from the same area.

The interior scans, they are in general very good. They are all colored but had to be subsampled, because otherwise most computers would not be able to open such an amount of data all at once. The clouds have been subsampled to 1 point every 2mm so a maximum of 25 points per square centimeter.

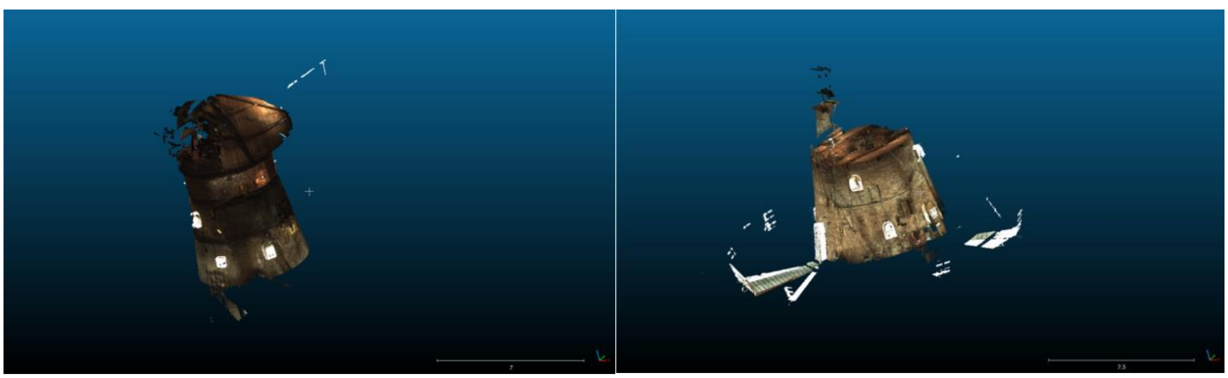


Figure 27 & 28: top 3 floors and 4th floor interior scans (resp. 54.4 and 29.6 million points)

The only funny thing in the scans is that a person has been captured in the scans two times, once on the balcony scans; which was quite easily to filter out. The other occurrence showed the leg of one of the operators captured in the color, and her fore much harder to filter out correctly, as it is embedded in the color shown below (a minor thing).

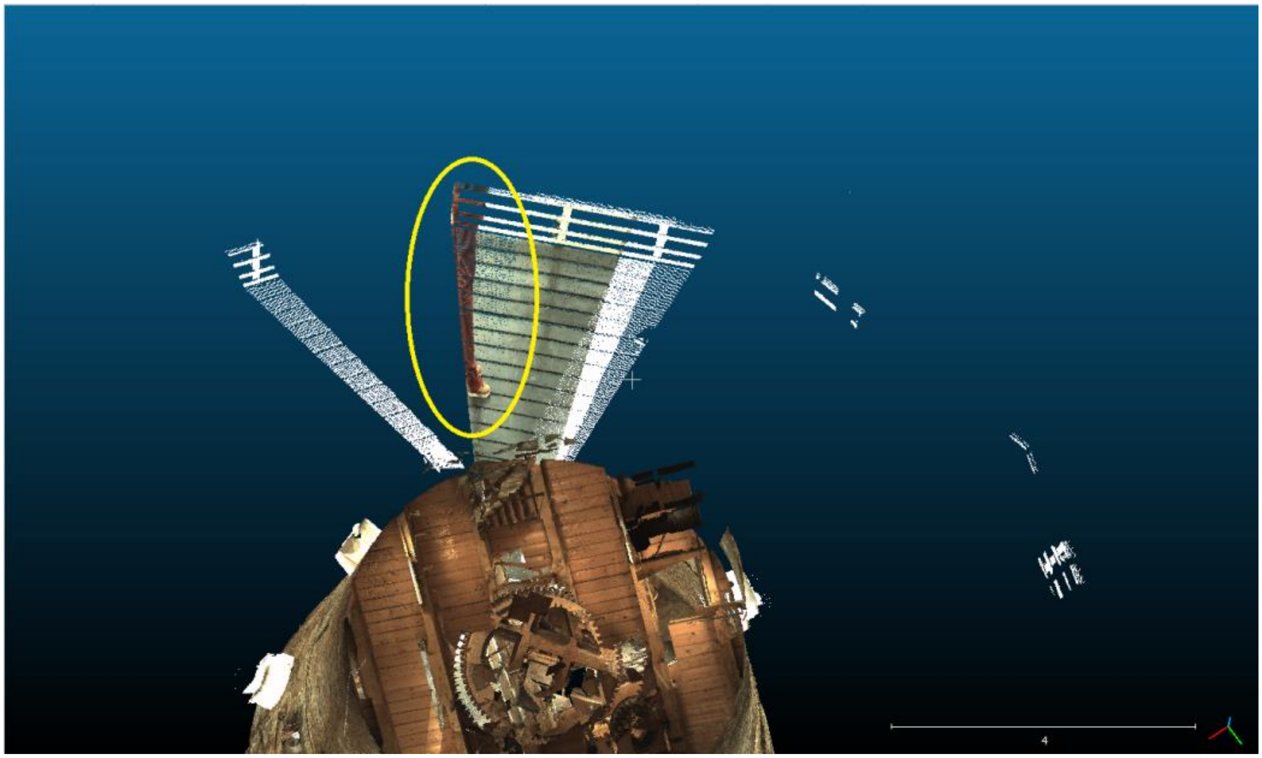


Figure 29: oops, feet captured at photo

Combining all these pictures resulted in the following point cloud of the whole environment, colored as good as possible and as complete as possible. The complete point cloud is shown on the next page .

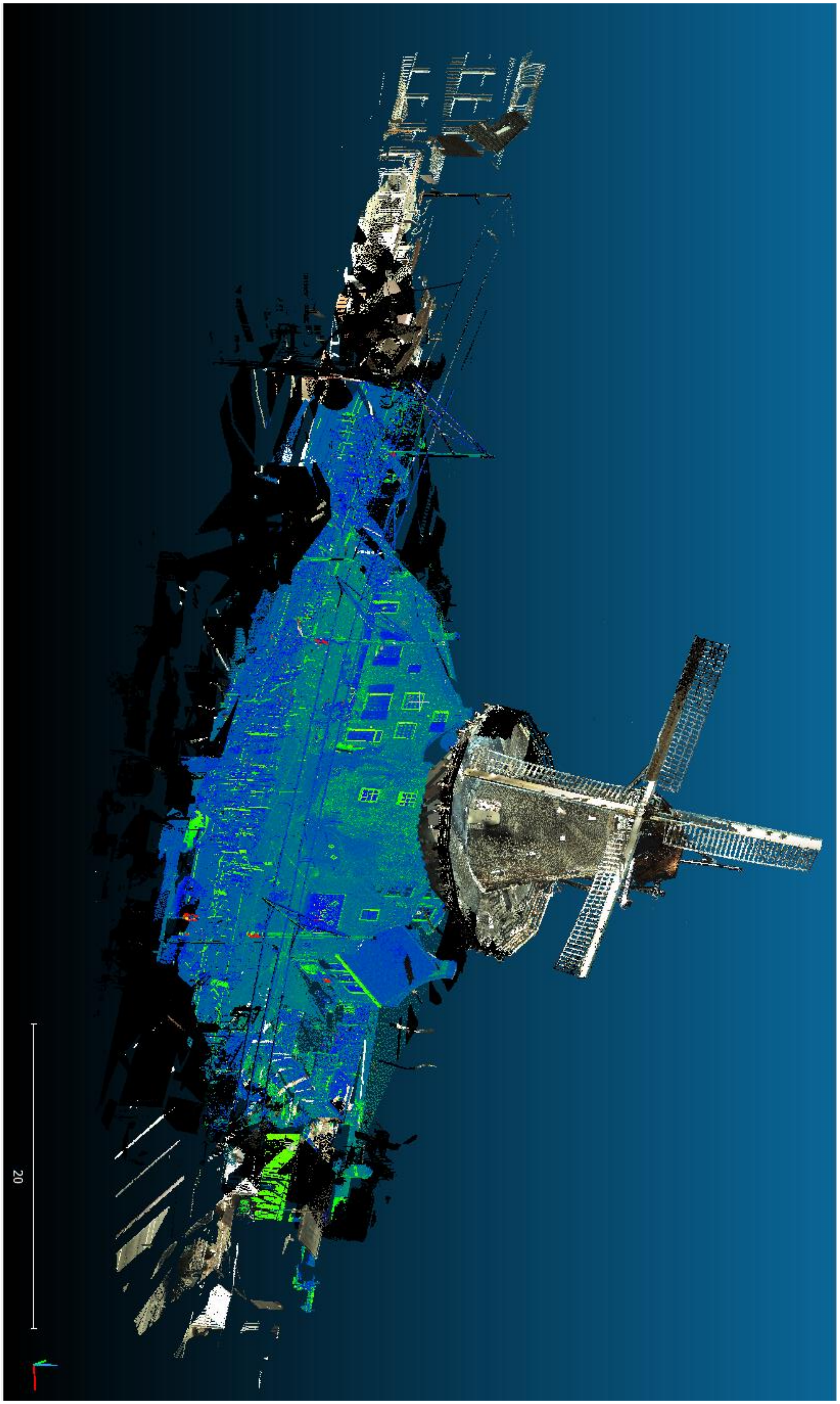


Figure 30: total merged exterior and interior scans, 124 million points.

3.3Meshing the point cloud.

3.3.1 Meshing in Cloudcompare

After merging all clouds together (over 124 million points) it is time to look whether it is possible to consider the meshing process. Meshing can be done in both Cloudcompare by the 2.5D Delaunay triangulation or in Meshlab performing Ball-pivoting or Poisson’s approach. Cloudcompare can handle as many points as possible, Meshlab has a point restriction. Also Cloudcompare’s computing time is more or less linear with file size while Meshlab’s computing time is increasing almost quadratic.

First, the hole building was cut in 3 pieces again, a part containing all interior floors, a part containing the roof of the mill and the blades of the mill and last but not least the exterior of the building with exclusion of the blades and the roof of the building. Later also the exterior was cut again into a part including the balcony, and the rest of the building separately ((will be shown later). The cutting in pieces was done in order to make the computation faster for the programs.

Below you can see the point clouds with which Delaunay triangulation has been performed in Cloudcompare.

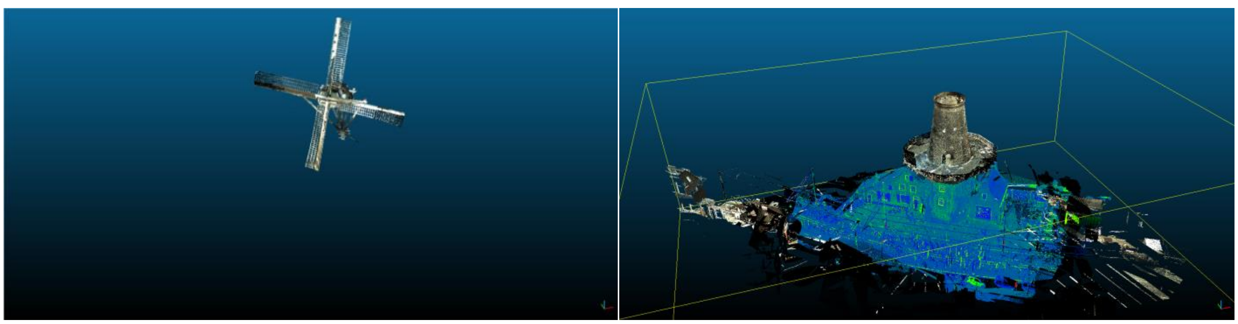


Figure 31 and 32: exterior point clouds used for Delaunay triangulation, 4.8mil and 37 million points respectively.

The interior cloud, which was equal to 83 million points, has been cut into pieces in order to make it faster to compute as well. The interior of the mill was cut into 5 parts, defined by the floors; every floor was cut separately (from the balcony floor up) except the top floor. The top floor was cut into two parts. The different cuts are shown below (very short and in small figures).

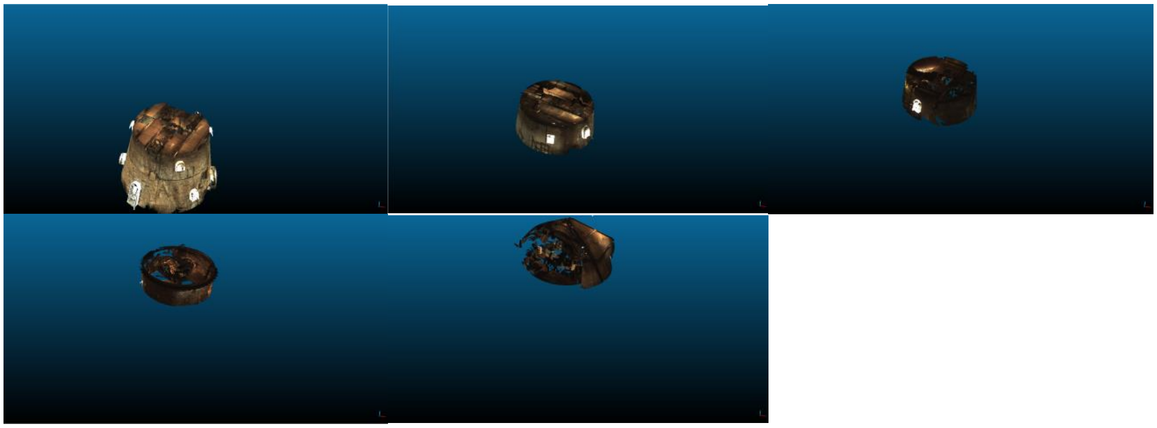
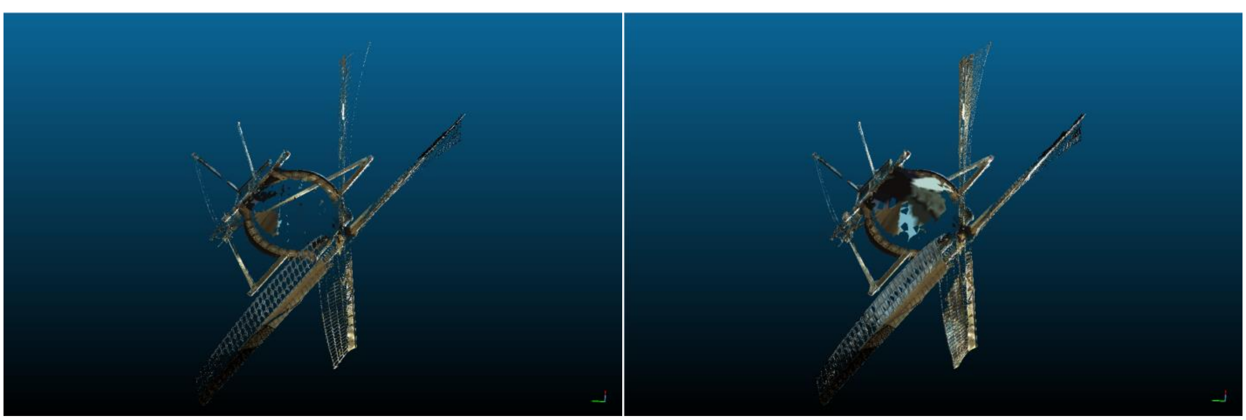


Figure 33: cutting of the interior part

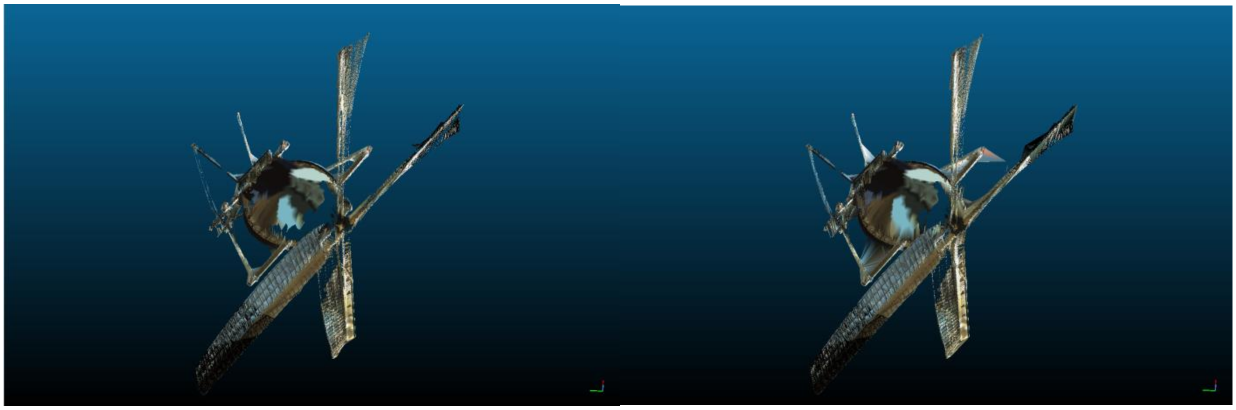
The different parts above are just a breakdown from the total merged interior point cloud shown at figure 27 & 28. With these point clouds, Delaunay triangulation has been performed. The triangulation has been performed with varying maximum edge length. The different clouds have been triangulated at original size and subsampled size (subsampling to about 500.000 points per cloud, done by subsampling the original cloud to a minimum distance of 0.010-0.015 meters between points) in order to look if there are differences between the different edge lengths.

First the results for the roof and blades of the mill are discussed. Every cloud in our database has been processed with maximum edge lengths of: 0.2meters (0.2m), 0.5meters (0.5m), 1meter (1m) and 2 meters (2m). All of the different meshes made from the roof & blades will be shown once, since it is good to see what the results are. However pictures of every mesh are not provided in this document since that are over 53 different meshes; just too much to show here. Only the important meshes which properly illustrate the differences will be shown.

Below are all the Delaunay meshes of the original point cloud of the roof and blades.



a b

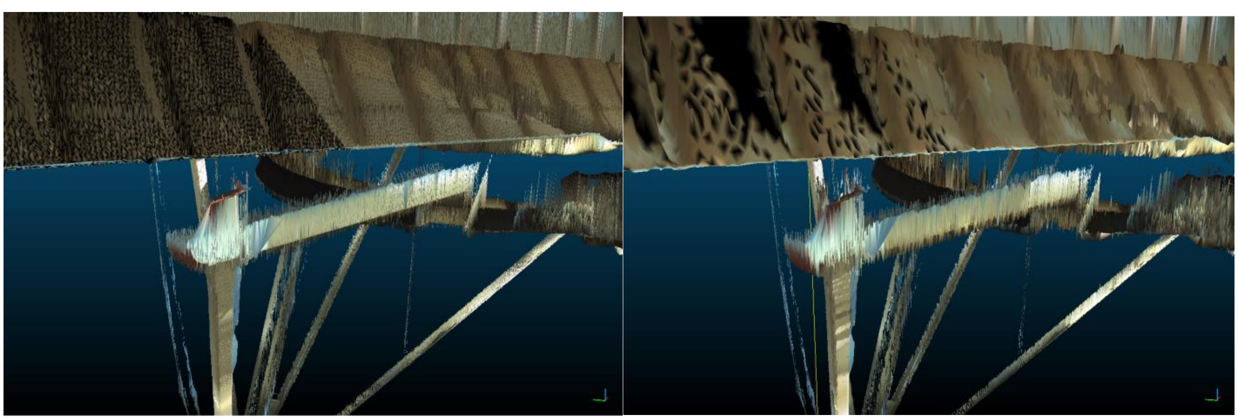


c d

Figure 34: a,b,c and d. respectively 0.2m mesh, 0.5m mesh, 1m mesh and 2m mesh.

Figures 34a to 34d show that as soon as the length of the maximum edge gets bigger, points which are further located from each other will be connected. The 0.5 and 1m meshes are still reasonable, but the 2m maximum edge length makes triangles which are not realistic. Of course meshes with different lengths between 0.5meters and 2 meters can be computed in order to see what the optimum is, but that would create a database which is too large to handle efficient and fast. Within our computed possibilities the 0.5m and the 1m mesh give the best possible Delaunay meshes for the roof and the blades. However, the pictures show that there are a lot of holes, especially in the roof of the building. It is concluded from the meshing that this method is very sensitive for point density (which will be discussed further below).

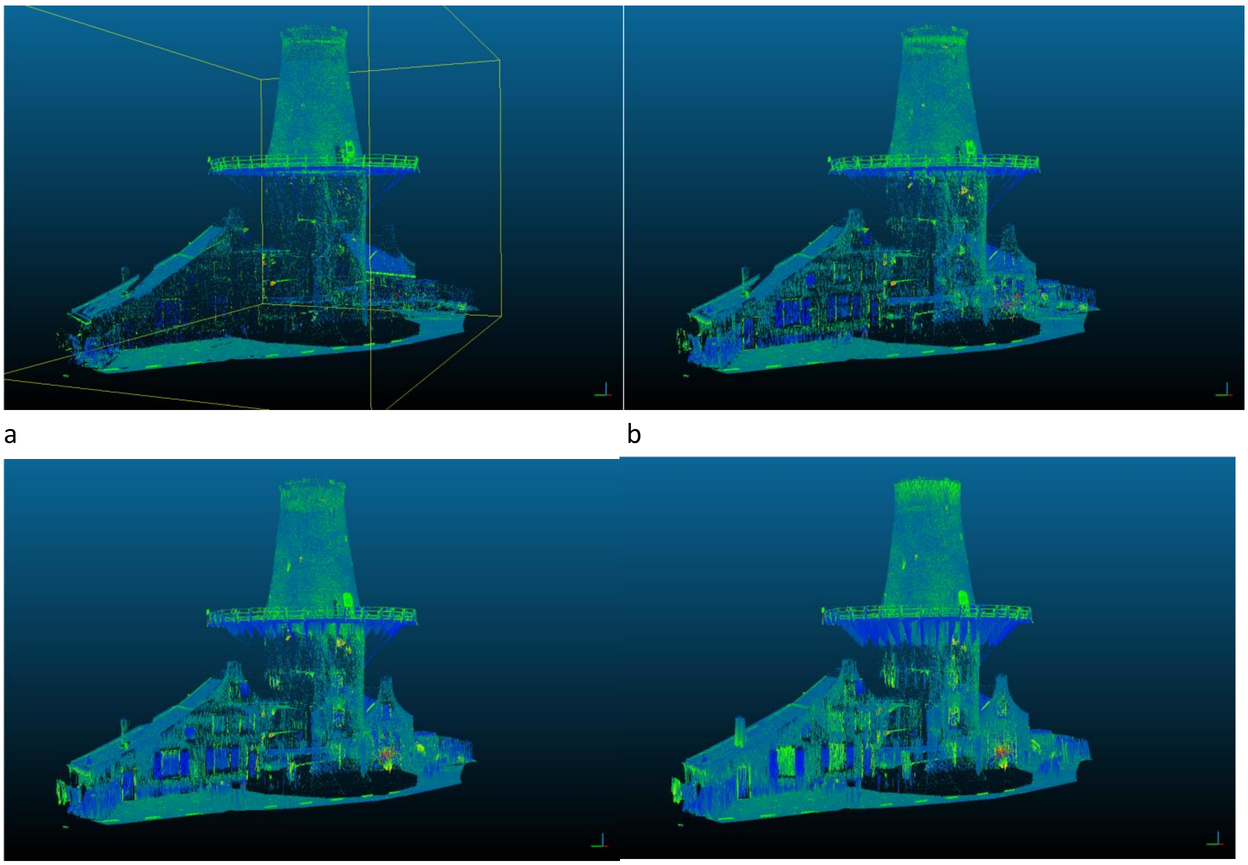
The data has also been subsampled (1 point every 0.015m) in order to make it computable for Meshlab, but these subsampled point clouds have also been meshed in Cloudcompare in order to see the differences. The result from down sampling is that the level of detail is deteriorating quickly as shown in figure 35.



a b

Figure 35: original and downs sampled point cloud meshes of the 0.5m mesh generation.

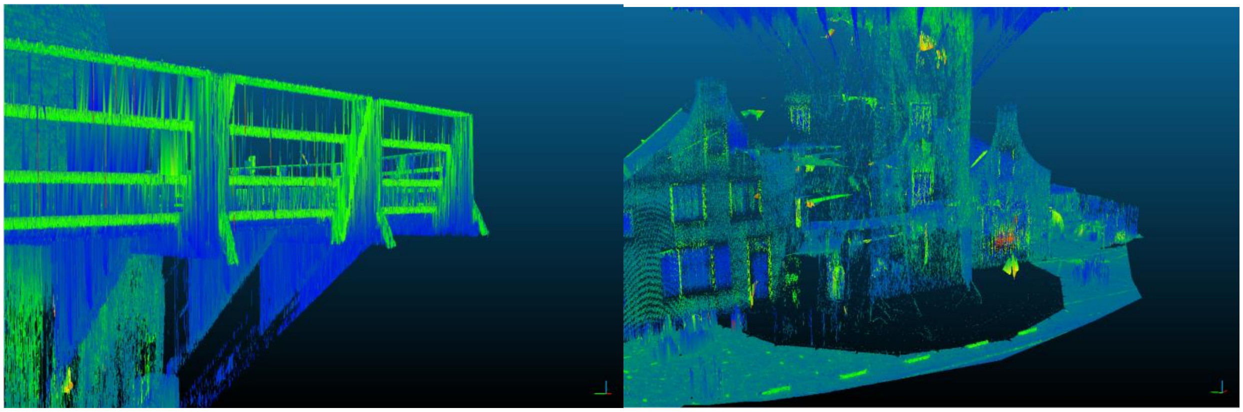
The down sampling results in faster computation, but also in loss of detail. With lower point density and different shapes visible in your point clouds, you have to be more careful with picking the maximum length of your edges since it is less clear (for the meshing program) which points belong to each other. The next topic is to discuss the rest of the exterior of the building. The total amount of points in the original cloud was about 30 million. The subsampling which has been performed was done with a maximum of 1 point per 0.05meters (in order to get the right amount of points). The results from the original cloud is shown below in figure 36 indicates that the Delaunay triangulation tends to triangulate vertical.



a b c d

Figure 36: a, b, c and d. respectively 0.2m, 0.5m, 1m and 1m mesh original cloud (30 million points)

As can be seen from figure 32, more points from the point cloud are taken into account when meshing (or the triangulation goes better) as soon as the maximum edge length gets larger (x m mesh). But again like for the roof meshing, also points are being connected which actually do not belong to each other. The vertical tendency can be found also for the balcony part shown below.

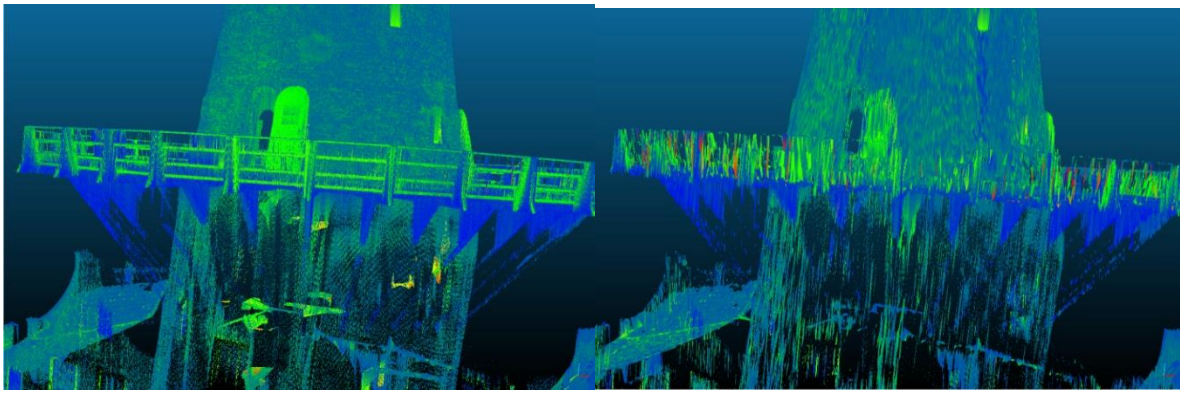


a b

Figure 37a and b: close-up 1m mesh balcony and close-up ground beneath balcony

In figure 37 is shown that the interpolation tries to connect the separate beams (clearly shown) with triangles to other beams, this phenomena occurs because the beams are still in the same x-y surface (when projected on an x-y plane). The distance between the beams still causes the triangulation to make a good distinction between the separate beams. Also in front of the mill, where there are enough points to make a mesh, the mesh generation throws out all points and leaves the area largely untouched. This area which is largely untouched is kept untouched because of the balcony straight above it. These two surfaces are more or less parallel and are therefore seen as belonging together (for the Delaunay triangulation). When making triangles the program sees that the distance between the two surfaces is much larger than 1meter, so the area directly below the balcony is not being triangulated. This vertical tendency and exclusion is one of the reasons why Delaunay triangulation is not very well suitable for this building shape. The 1m distance triangulation is however the best triangulation. Triangulations with lower edge length are excluding too much data while the 2m edge triangulation makes triangles which are somewhat unrealistic.

Also the exterior has been subsampled down to a more reasonable size. Below the difference with respect to meshing is indicated. The sample density is a maximum of 1 point per 0.05m (at the down sampled point cloud).

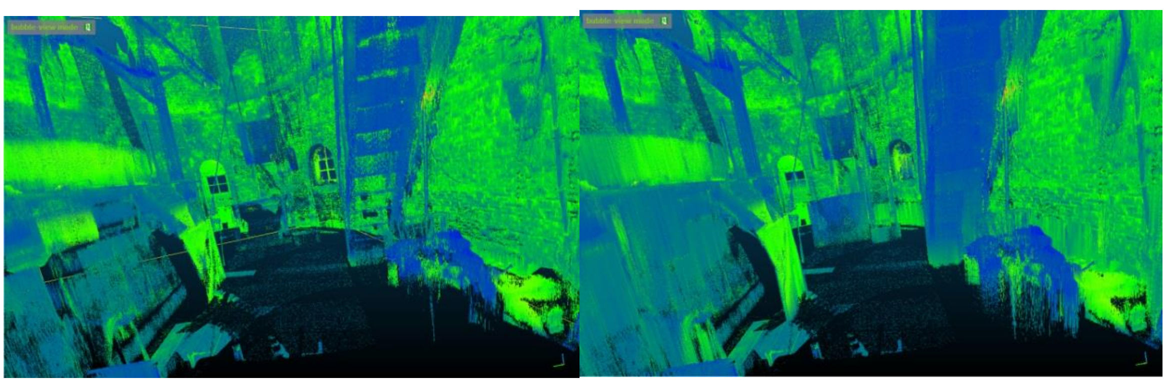


a b

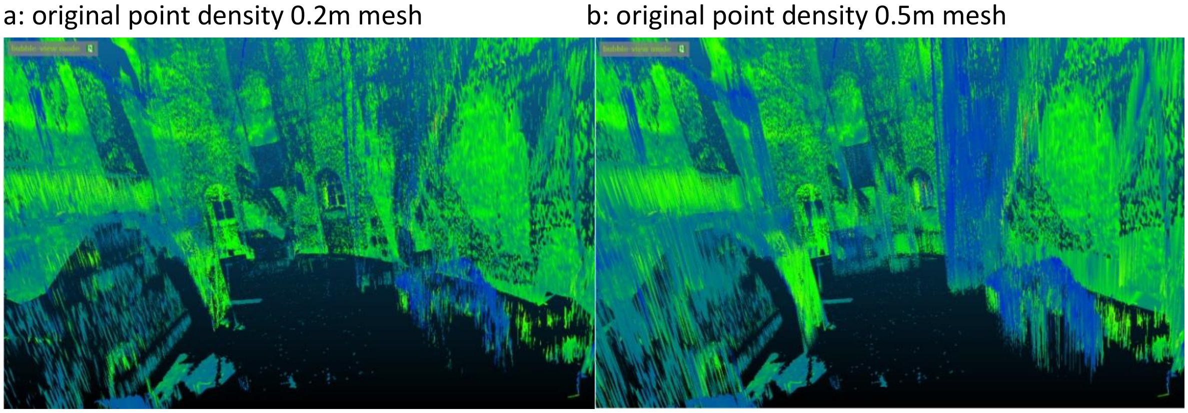
Figure 38: original point density mesh and down sampled point density mesh

Figure 38 indicates that the down sampled point density makes a great different at the balcony. There are many irregularities on the windmill, on the balcony and on the body of the windmill. Therefore it is studied later if the other triangulation methods work better.

Finally, the interior of the building is considered below are 4 pictures showing the 4th floor (balcony floor) in the original file size and the subsampled point density (to 1 point every 0.015m). The meshes are shown for a maximum edge length of 0.2 and 0.5meters.



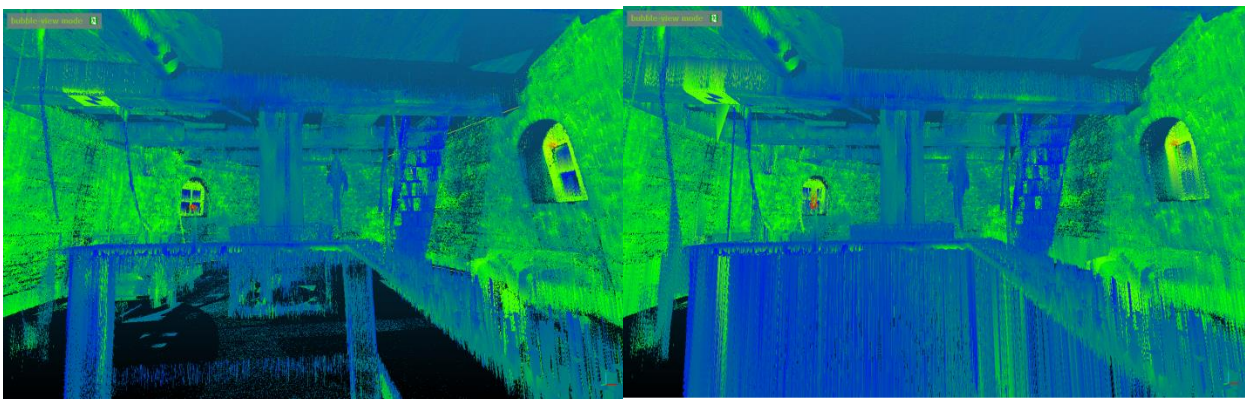
a: original point density 0.2m mesh b: original point density 0.5m mesh



c: subsampled 0.2m mesh d: subsampled 0.5m mesh

Figure 39: 4th floor interior meshes

Shown is that the point density is more attractive when being large. The triangles get very irregular when the point density is lowering. Figure 39 indicates that the work has been performed on a smaller scale. Because of the smaller point density and the smaller working area the triangles get irregular and easily become unpredictable. Only the meshes of 0.2 and 0.5meters maximum length have been shown because any larger lengths would be unrealistic and would result in an incorrect triangulation. Figure 35 till figure 39 all show that the original point cloud is much better than the subsampled point cloud. The same evaluation can be done for the other floors and result are similar with respect to point density and maximum edge length. Below the 3rd floor is shown. Images of the 2nd and 1st floors are included in the appendix.



a: 3rd floor mesh original density 0.2m

b: 3rd floor mesh original density 0.5m

Figure 40: 3rd floor meshes

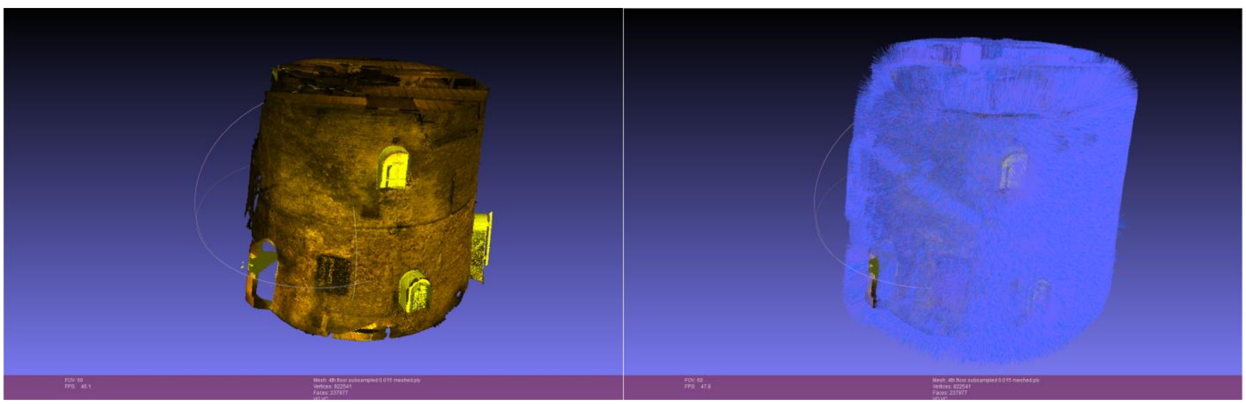
Observed is that the shapes of the objects are there, only the floor is excluded (same reason as for the balcony part) because their surfaces are parallel, the program thinks they belong to each other, but since the distance tween the planes is too large there are no triangles formed, and the points are deleted.

The remaining pictures of the remaining floors are found in Appendix I.

3.3.2 Meshing in Meshlab

Chapter 3.2.1 has learned that the Delaunay triangulation was not very successful at low resolution, both with respect to surface coverage and as being a good representation of the reality, since the method tends to make vertical triangles. Different methods with a different program are now considered, in order to look if other methods provide different and better results. The two methods which will be computed are the ball pivoting surface reconstruction and the Poisson Surface Reconstruction. For both of these methods, normals need to be computed. These normals are constructed by looking at adjacent points of the point from which you want the normal. By considering the relative position of these adjacent points, their normal vector can be determined. Important to be noted, is that the total cloud is cut into 10 parts: 4 balcony parts, 5 floors interior and the remaining of the exterior below the balcony. These parts are individually down sampled to about 500.000 points per cloud in order to keep the computing time reasonable. The down sampling also reduces the level of detail!

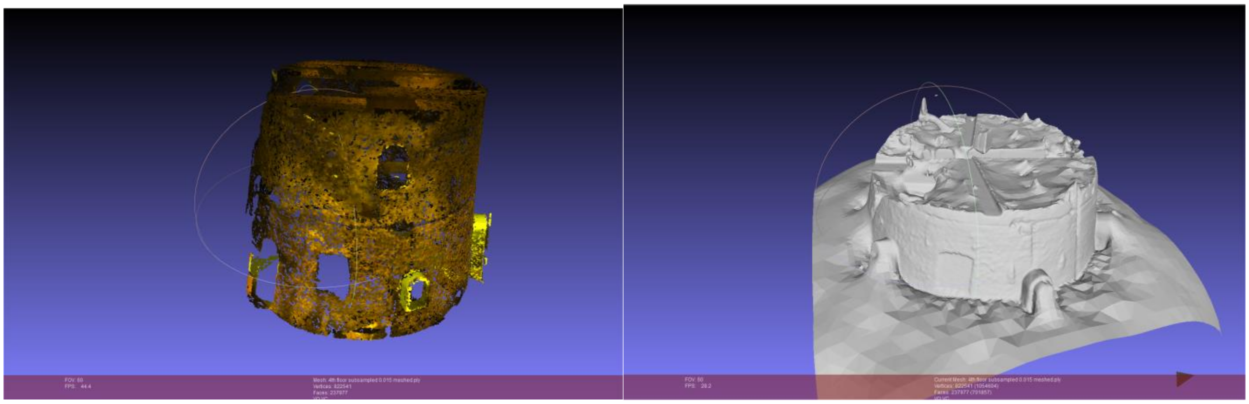
Figure 41 shows an example of how the normal field looks like, in comparison with the normal points on the 4th floor, the balcony floor.



a b

Figure 41: points of a part of the mill in Meshlab and their respective normal.

Figure 41 shows that for every point in the point cloud, a normal is computed**.** These normals are now used to perform both ball pivoting and apply the Poisson reconstruction algorithm. Both the algorithms have their strong points and their weak points, shown in figure 42.

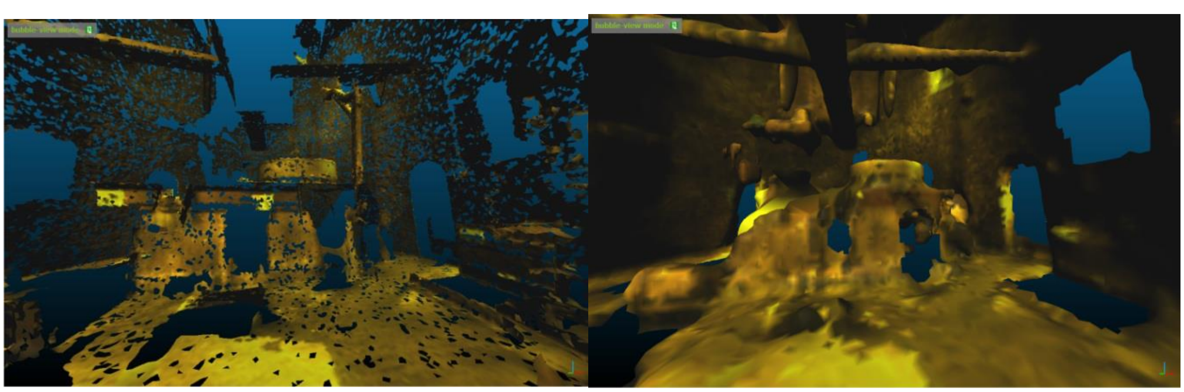


a b

Figure 42: ball pivoting and Poisson reconstruction.

The ball pivoting reconstruction (left) gives a good representation of the surface of the walls and some of the interiors of the building. However, its drawback is that the method makes use of balls with different radius in order to perform the pivoting as shown in the method. These balls are not suitable for the complete point cloud because of different point densities and angles present in the cloud. Therefore holes appear in the model where the pivoting ball could not properly construct triangles. The Poisson reconstruction on its turn is attractive because it is creating a complete surface, without any holes. However the right side of figure 42 also shows that a part of the floor is missing. The algorithm does not always fit a surface through all of the points. So also with this algorithm one cannot map all of the insides of the windmill simply because the algorithm cannot fit a sufficient representative Poisson model through the points. The Poisson method also creates balloon like surfaces, the surface tends to be closed like a balloon, and the method does not create 90 degrees angles.

In order to obtain a suitable mesh, parts of the different meshes need to be cut-off. The ball pivoting has many holes in it and some parts of the insides are generally excluded, while the Poisson method makes a full surface with no holes, but sometimes makes strange surfaces which needs to be cut off in order to be more representative. In the figure below, the example for the two meshes are shown for the 4th floor.



a b

Figure 43: ball pivoting and Poisson reconstruction 4th floor interior

The Ball pivoting method creates a surface which is close to the correct model, however not complete and containing holes. The Poisson reconstruction method creates a complete mesh, but with strange bubbles apparently without any purpose. In figure 43 b, the Poisson reconstruction mesh has already been cut to only represent the most realistic parts. After this step both meshes are merged to one mesh in order to make the most complete representation. In the figure below, the merged result is shown.

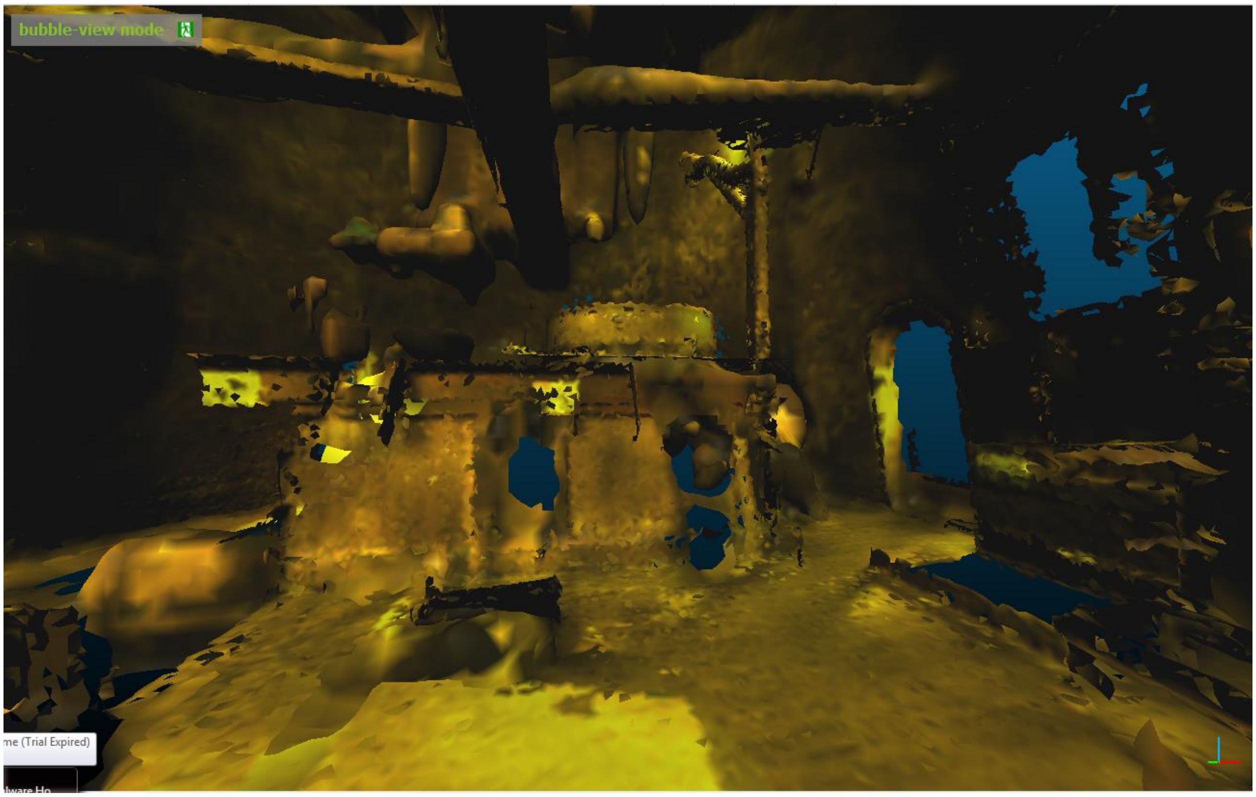
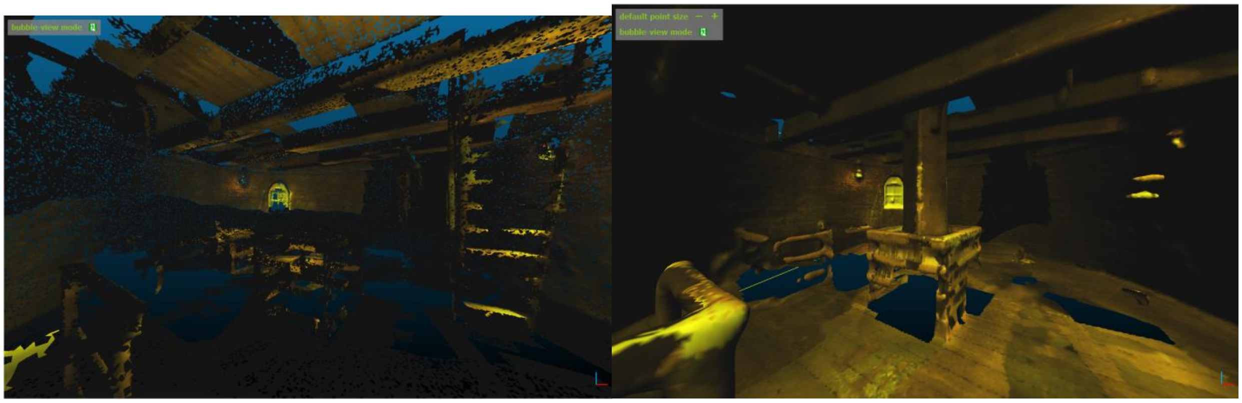


Figure 44: merged meshes.

Just merging the clouds is not satisfactory to make a complete mesh, since the meshes will create an overlay (two layers of which only one is seen) representing the same surface. The better solution is that parts of the clouds are being merged together, so that there is only one surface present representing one surface in reality. This is however, not a problem which can be solved within this project considering the available time. Therefore the 3D model will be represented as shown above. The 4th floor is one of the messiest floors present in the model. Other parts of the mill (with exception of the top floor) are a lot better. Below one more result is shown of the two clouds separated and merged, floor 3. The remaining results of the different clouds can be found in Appendix 2.



a b

Figure 45: Ball pivoting and Poisson reconstruction 3rd floor.

Figure 45 indicates the many similarities between the different mesh generations. The roof floor and walls are very good (with exception of holes in the ball pivoting method). The Poisson method actually creates a very good representation of the building with exception of some parts as shown in figure 45 b, where the bad parts have already been cut out. The combined mesh created from the images in figures 45a and -b is shown below.

Figure 46: combined mesh ball pivoting and Poisson generation

Again there are still some remaining holes in the meshes, but that is because the meshing methods are not perfect for the encountered shapes.

The results from the meshing are in general good, but the different meshing methods encounter several problems when combining different shapes or different point densities. Although most point clouds have been subsampled so that the point density is as regular as possible, (distance between points mostly 0.01-0.015 meters) still the meshing was not satisfactory. In the future, it could be a possibility to combine the three considered methods. All of these methods have their own strengths and weaknesses. Ball pivoting creates a very accurate surface (when the triangles are being created). The Poisson method is giving a closed mesh at all times, but not really a representative surface at all times. And the Delaunay 2.5D triangulation (which is related to the Poisson method) is very accurate at recognizing top surfaces of objects. Further recommendations on how to combine these different triangulation methods are provided in the discussion in the next chapter.

The visualization of the other results from the exterior, the balcony and the remaining floors, are shown in appendix II.

Below is the total mesh shown, both exterior and interior are rendered there (the exterior is better visible than the interior).

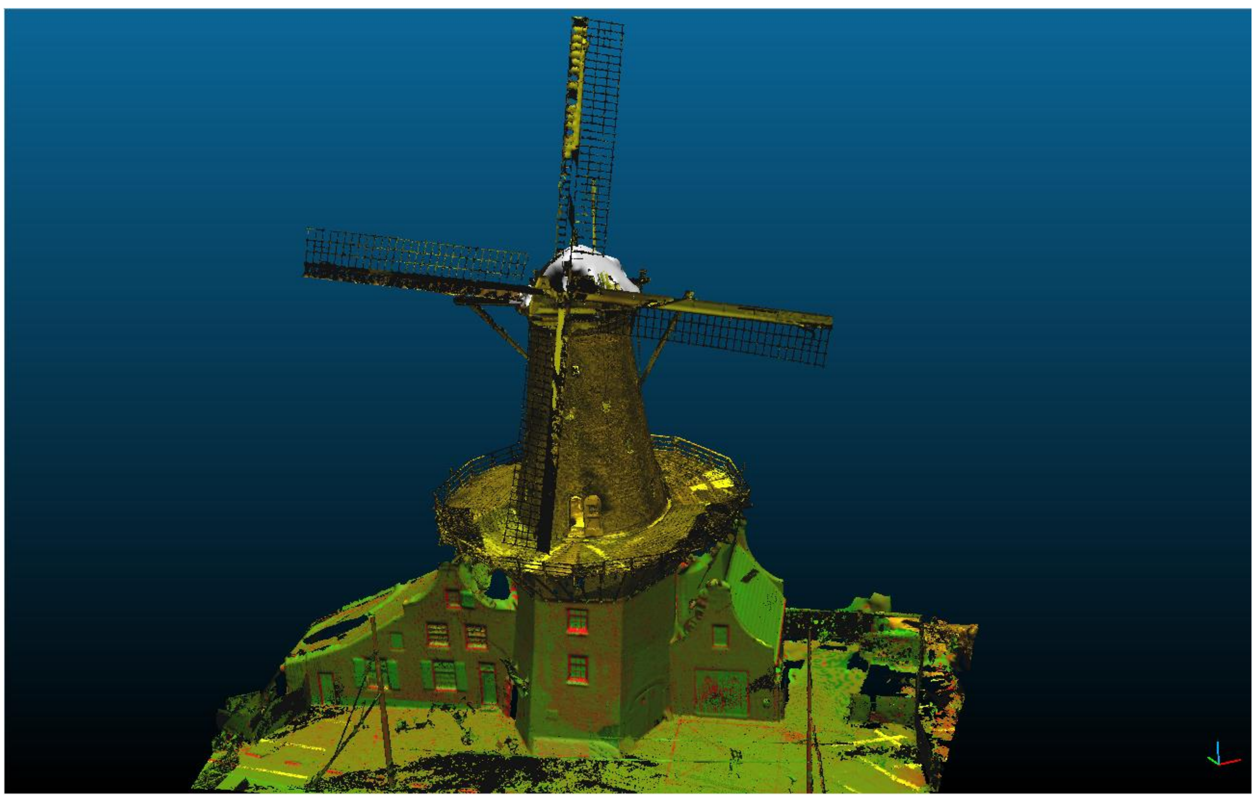


Figure 47: total merge exterior and interior

In this stage of the report it is good to recognize that not all of the Poisson meshes were useful. The Poisson mesh generated for the balcony (fig 59b) was not useful at all, since it was very irregular. The cut from the balcony shown in appendix II shows that the Poisson mesh is not very good, and therefore it has been excluded from the results.

4. Discussion.

For future calculations on accurate dense sampled point clouds, there are several recommendations which could be useful when processing the clouds. These recommendations do not only include the ways to process the data, but also includes computers and advices for programming.

First of all a recommendation is made on the general processing order at the start of the work. A significant amount of data has to be processed, irregularly spaced data, colored and uncolored. There is a significant amount of noise in the data and it is often not aligned very well.

4.1 Way of processing data.

From the analysis in this report a procedure to handle the large amount of data and processing it, the following is recommended:

1. Segment the different point clouds, taking care of excluding data which is outside the scope of the project.
2. When the data is not registered, the next step would be to try and finely register the data so that all the different scans will eventually be aligned properly and all represent the same object in space consider linking only parts at once, not everything at once.
3. When dealing with very large datasets which are aligned it might be useful to first subsample the data in order to prevent the danger from overloading the computer with respect to its RAM memory.
4. When scans from different sources are being used, where sometimes colors are included and sometimes colors are not included. It is nice if color is being interpolated from other clouds in order to get one uniform colored model. The result will be one model where every pixel has a color.
5. After all the clouds have a color, the merging of the clouds to one big cloud can start. Care must be taken that it is possible to open the individual clouds at once. If this is not possible, there is insufficient ram available or the point density is too high. Another issue is that the graphics card has insufficient specifications.
6. After merging it is advised to first perform the segmentation In order to define different objects to mesh.
7. After segmentation, meshing has to be performed. This takes considerable computing time, especially for the very large clouds (> 100million points). Different meshing methods can be used in order to generate the model, using different programs.

It has been mentioned that that none of the meshes were perfect. A method is needed to combine the strong parts of the different methods. From the above analysis it is concluded that if we would be able to combine the strong parts of different methods, a more complete model would be created better than the result shown in this document. For example, the 2.5D Delaunay triangulation does the top surfaces of objects very well. Most of the top surfaces of objects are being recognized. However walls are very poorly done by 2.5D Delaunay triangulations. The walls on their turn are very well done by the ball pivoting surface reconstruction. However, the ball pivoting reconstruction has troubles with this dataset somehow; something which should be improved resulting in better coping with these amounts of data with providing better mesh coverage. The meshing density is very well done by the Poisson surface reconstruction algorithm. So the combination of the accuracy of the ball pivoting with the meshing consistency of the Poisson meshing method would create a very good model.

4.2 Computer requirements.

Datasets are very large. A simple method to open this amount of data must be searched for. It is impossible to open 250million points in color, with only 8 GB RAM memory. There should be a smart way of calling data without overloading your memory, even for an only 8GB system. For 16 GB or 32GB RAM, 250million points or even 500million points should not be a problem. In that case the video card will likely become the bottleneck. The rendering of points does not only depend on RAM or CPU, also the graphics card needs to be able to do so. Some graphics cards are just too low end in order to show the amount of data you want to show. So all together it depends on actually all of the components of a computer: CPU, GPU (graphics card) and RAM memory. So there should be a smarter way to deal with the large data sets, for instance to build in a program creating easy access through a different memory location , to easier access large amounts of data and load it into your RAM memory more efficiently.

4.3 Recommendations regarding programs.

The used program, Cloudcompare, has an easy usable interface, which is very clear. But sometimes it seems like the programming is not consistent with respect to CPU usage. Sometimes the program only uses 15% of the total CPU; while the computing takes 15 minutes (this is the case with statistical outlier removal). So this time could be reduced if the program uses more of the available CPU. One would expect that this depends on the computer. Apparently the program is not allowed to use all available CPU. This issue does not occur for other computations such as color interpolations by other entities. In that case the CPU is pulled to its maximum of 100%. Obviously in that case the computer is not available for other tasks. There should be value defining how much power of your CPU can be allowed for a given task. Maybe this is already embedded, but such input is not found. This recommendation is not only useful for low end computers but also high end machines functioning as a work station at the same time.

4.4 Registering.

There are 2 popular methods for data registering. The first one is ICP (iterative Closest Point). This is an algorithm which makes use of two point clouds. These point clouds are being aligned iteratively by picking point within the clouds which are common points. This point picking is not done by hand but by an algorithm. Every step that is made, the algorithm tries to bring the clouds closer to each other. The other popular method is registering with features. These features include spheres, planes, points, cylinders and more. Here the features are being matched by giving weights to the defined features, and also iteratively the features are being aligned. Somehow these methods should be combined so that IPC would be performed with the registering with features. Registering with features is not able to rotate an image while the IPC can still do this (slightly).

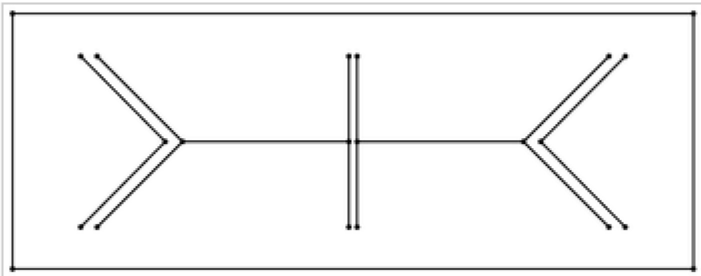
For a full explanation is referred to the article under reference (10)

4.5 Mesh generation (Triangulation).

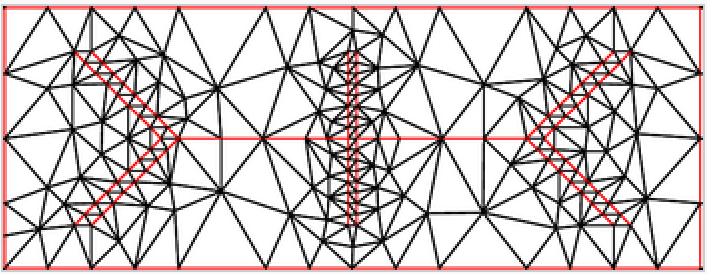
Many different types of triangulations possible. This project tried and explored the 2.5D Delaunay triangulation and Ball pivoting triangulation. Of course also the 2.5D best fitting plane model could be used in order to perform the mesh generation model in Cloudcompare. But that is only possible when the surfaces that are meshed are perfectly flat and simple, because the algorithm will fit the mesh with a flat plane.

Other meshing algorithms in Meshlab can be considered as well. Here only the ball pivoting algorithm has been used. For surface reconstruction the Poisson method has also been used, but the results coming from that method are way too balloon-ish (blades from the mill look like balloons when the surface is being constructed). The results from The Poisson method have been used to complete the ball pivoting mesh. One method worth considering is the surface reconstruction called VCG approach. It is a relatively old method, which has been used for a long time. The full explanation for this method can be found online. Another interesting method for the mesh generation is the Fortune's algorithm. It is an algorithm which produces the vonoroi diagram of a point cloud in a sweep like manner (moving from one side to the other). This could possibly be interesting when looking at methods which make use of the vonoroi diagram with respect to efficiency or speed of the mesh generation.

A variation on the Delaunay triangulation is Ruppert's algorithm. This algorithm also known as Delaunay refinement is a better version of the Delaunay triangulation (it has a better quality).



Input planar straight-line graph



Output conforming Delaunay triangulation

Example of Ruppert's algorithm

Figure 48: Example Ruppert's algorithm (14)

There are several other option that could be considered, such as Tessellation and K-D tree. Both have been brought to live in order to make rendering structures easier by "cutting" the data in pieces for more area specific rendering. For future processing it could be useful looking at direct computation with Matlab or Python for the ball-pivoting algorithm. The algorithm takes some 1 hour to compute for 500k points while the article under ref. (12) describes a computing time of not more than 2 minutes for a Buddha point cloud containing 3.3 million points. This means much computation power has been lost somewhere in Meshlab, although it remains (for the author of this document) unclear what causes this loss.

4.6 Future possibilities.

More options than only the improved current acquisition methods will be available in the future. AHN data acquired by the Dutch government is still an important option with respect to roofs (information on this tool can be found in reference 15 and reference 20). This technique has been used for this project because it provides the possibility of adding information with respect to roofs, which cannot be scanned. But AHN is very costly and the data density is not that great. Therefore it might be interesting to also consider new technologies such as drone scanning (AUV). This technique is still under development but it could provide very accurate and dense data of areas which are hard to reach for example roofs of houses, churches and windmills. Ground techniques have been significantly developed lately, but those techniques result in very scarce data density of roofs. The only problem with the use of drones would be that according to Dutch law, a permit to fly and captures images is needed. It would also be an idea considering capturing data points with smartphones. In this project the data processor concluded many times that data was missing (shadows) because the scanner could just not reach some position. It would be very useful if in a later stage some parts could be scanned again, for locations that showed missing information in the original scans. Mobile phones are as the name says much more mobile, therefore would be easier to capture point clouds on places you cannot reach with a regular laser scanner). These relatively new possibilities are worth looking at, since they will make the life of the processer (a person) much easier.

Large datasets are accessible for trained people. They have a trained mind and hand in certain programs for visualizing certain results (in this case point clouds or meshes). It would be a good idea to look at what is called “WebGL”. WebGL is a way to visualize 3D meshes (generated with complicated programs) in a simple way. WebGL is easy to handle for almost everyone because the controls are just your mouse. Below is an example of a visualized mesh in WebGL made in by Researcher and Supervisor of this bachelor’s graduation project Beril Sirmacek (see reference (16)). The visualizer is operable on almost every browser on every platform. Performance of the WebGL extension for the individual browsers on platforms can be found under reference (17).

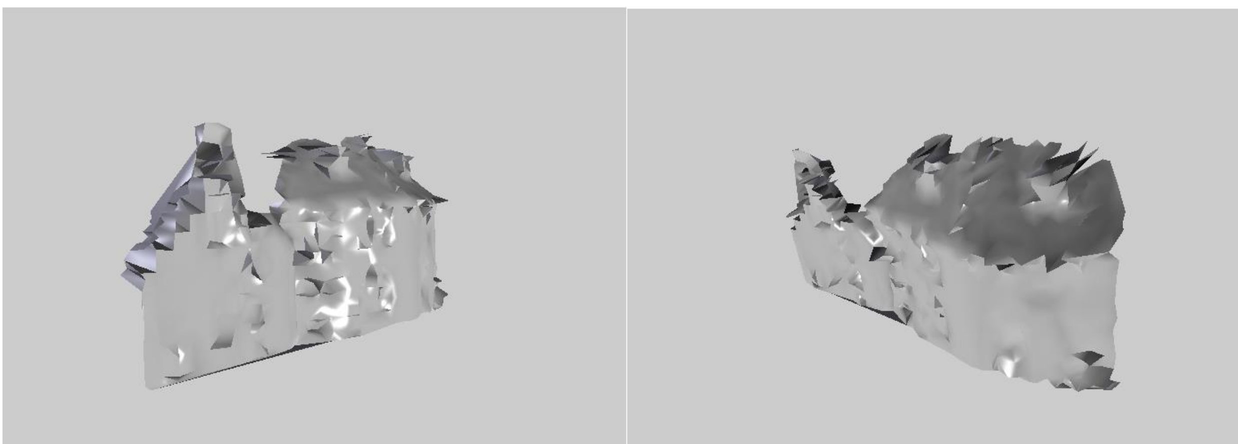


Figure 49: WebGL showing a small test of visualizing a mesh

Data gained from processing, the point clouds and meshes, can be used for non- scientific purposes as well. The results can be used in order to make videogames, or animations or a virtual walk through game. Such a program is called “Maya”. The reference is found under (19); with this program animations, modeling, simulations and rendering can be performed.

5. Personal Experiences.

This is the chapter where the more personal experiences of me, the author, are being shared. There are a lot of different things I encountered while working on this project. The personal experiences are subdivided into more practical experiences in the field and more processing experiences.

5.1 Practical experiences.

5.1 a: Moving the scanner.

There are a lot of procedures which went better than expected or were harder than expected. Some of this concerns the Leica c10 Scanstation. This Scanstation is rather expensive, so you don't want to break it. This machine is very sensitive to vibrations and changes in its position. The device is also very heavy. This caused me to be extremely careful with the scanner. Simple walking on pavements cannot be done, since the size and weight of the tiles combined with normal walking speed causes the device to vibrate according to its own natural frequency. Because of the weight of the scanner (+/-30kg), carrying it up the windmill was a heavy job because you were constantly trying to prevent it from bouncing into objects.

5.1 b: Linking scans.

When eventually the station has been set-up, you have to make sure the scanner is correctly referenced in space. In order to be very accurate we had to use many targets. Three targets is a point in space, so every scan must be linked to other scans with at least 3 targets. This was a real practical problem caused by all the beams and other wooden objects in the tiny spaces in the top of the windmill. Also linking every floor with at least three targets was quite a job. The scanner can only measure 45 degrees down from its horizon. The staircase openings are often only $1m^2$ in opening and often steeper than 45 degrees. Measuring down was generally not an option, if we wanted to link the floors we had to measure targets in one floor from the floor beneath since measuring down was often not an option.

5.1 c: Leveling the scanner.

During measuring care had to be taken not to stand in the scan (it still happened twice). Getting the scanner level was often a little bit of a problem since the floors are from wood, which is flexible. When walking away from the scanner, the floor would deform and the scanner would no longer be "horizontal" (with some maximum of, perhaps 0.5 degrees). But it appeared as very positive that the device was not influenced by the train driving under the building regularly. This while we, the operators, could often feel the vibrations from the train passing by underneath the building. The fact that it was not influenced by the train vibrations was caused by the significant weight of the scanner (very heavy stabilizer).

5.1 d: Scanning.

The scanning itself went very fast. It took approximately 5 minutes to scan from one point of view and about another 15 minutes to make photos. All together the scanning was not consuming the most time. Setting up the scanner was a similar effort. So the bottleneck is not scanning but setting up the scanner for scanning because it takes more time than expected. This could perhaps be solved when new techniques arise.

5.2 Processing experiences.

5.2 a: Exterior data.

The previous made scans were not very consistent. Only a few of the scans were aligned and every scan was from a different angle and with a different point density. All these exterior scans together were some 11 million points (not very much for the amount of area covered). But in the end all these scans were reasonably merged together with the AHN laser scan data (Sensor fusion).

5.2 b: Self-made scans.

Early during processing it was found that the amount of created data (in total 18 gigabyte) was too much to handle for a computer at once. Processing had to be done in parts, per floor or per two floors. Opening all data at once would cause most computers to crash (and also the one used in this study). Significantly more RAM memory is needed, for example 16GB or 32GB, in order to be able to open this amount of data. You could try to implement smarter methods with only partially loading data into the program or some tools allowing data loading directly from the hard disk. In addition better visualization should be generated by the graphics card.

The aligning of the scans was already done by Adriaan van Natijne in Cyclone, using a program provided with the Leica scanner. The method for aligning scans regarding the interior is much more precise than the linking of the exterior. The exterior scans did not have any targets, while the in this study used inside targets did have a millimeter precision.

In order to be able to process all the data at once, and to make it into a mesh in Meshlab, the point density had to be drastically downscaled from in 124 million points to about 5 million points (4% of the original). Much detail gets lost when doing this, but otherwise it is nearly impossible to process data within limited time.

Reference List

(1) http://forum.david-3d.com/viewtopic.php?t=4629
(2) http://www.geographica.hr/english/applied_geodesy/civil_engineering/tunnels
(3) http://skandrot.wix.com/laserscanningdunes#!terrestrial-laser-scanning/c14ij
(4) http://cipa.icomos.org/fileadmin/template/doc/KYOTO/14-2.pdf
(5) http://www.delftsemolen.nl/
(6) http://www.laser-scanning-buildings.co.uk/laser_scan_example_surveys.html
(7) Jinhu Wang, Ph.D candidate at the department of geoscience and Remote Sensing.
(8) https://www.kaggle.com/wiki/RootMeanSquaredError
(9) http://en.wikipedia.org/wiki/Standard_deviation
(10) http://www.isprs.org/proceedings/XXXVII/congress/5_pdf/105.pdf
(11) http://en.wikipedia.org/wiki/Delaunay_triangulation
(12) http://www.research.ibm.com/vistechnology/pdf/bpa_tvcg.pdf
(13) http://hds.leica-geosystems.com/downloads123/hds/hds/ScanStation%20C10/brochures-datasheet/Leica_ScanStation_C10_DS_en.pdf
(14) https://en.wikipedia.org/wiki/Ruppert%27s_algorithm
(15) http://3dsm.bk.tudelft.nl/matahn
(16) http://berilsirmacek.com/webgl/windmill-buildings.htm
(17) http://caniuse.com/#search=webgl
(18) http://research.microsoft.com/en-us/um/people/hoppe/poissonrecon.pdf
(19) http://www.autodesk.com/products/maya/overview
(20) http://www.ahn.nl/index.html

Appendix

Appendix I

Appendix 1 belongs to the results regarding the meshing in Cloudcompare with the 2.5D Delaunay triangulation. The subsampling indicated if it is the original point cloud or the subsampled point cloud where the point density is lowered to a minimum distance of 0.01 and 0.015meters between the points. The 0.2m and 0.5m indicate the maximum edge length of the triangulation. The larger the number, the larger the maximum distance over which a plane will extent. At 0.2m length this will mean a maximum plane length of 0.2 meters and at 0.5m this will mean a maximum of 0.5 meters span of the plane.

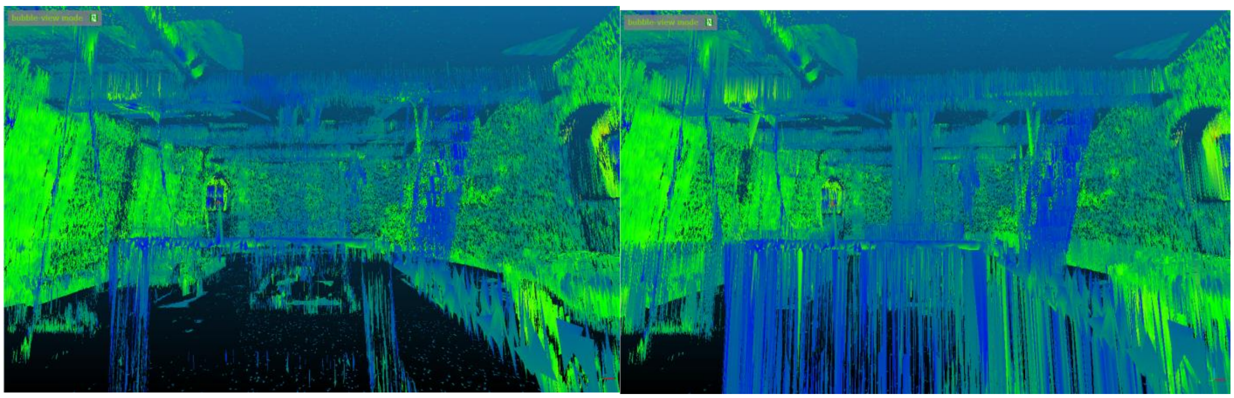


Figure 50: subsampled floor 3 with 0.2m and 0.5m mesh

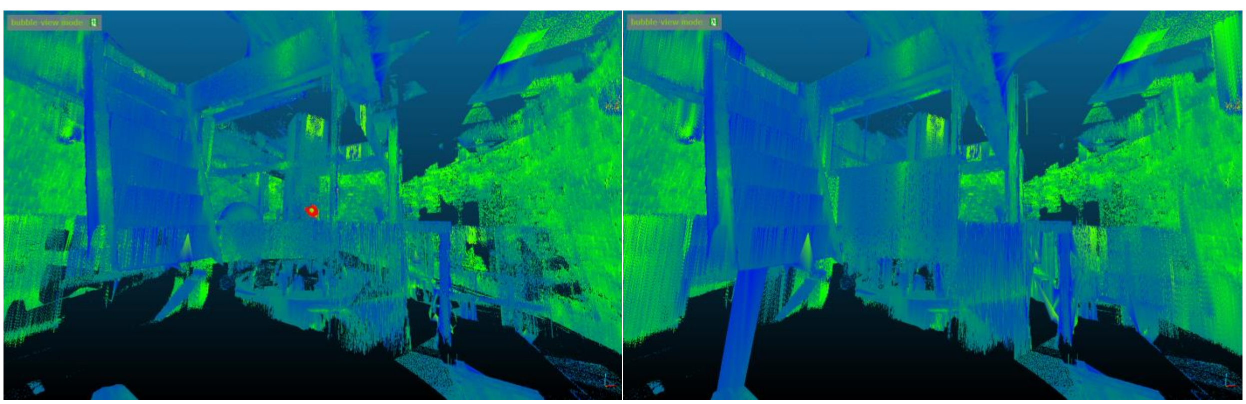


Figure 51: 2nd floor original mesh 0.2m and 0.5m mesh

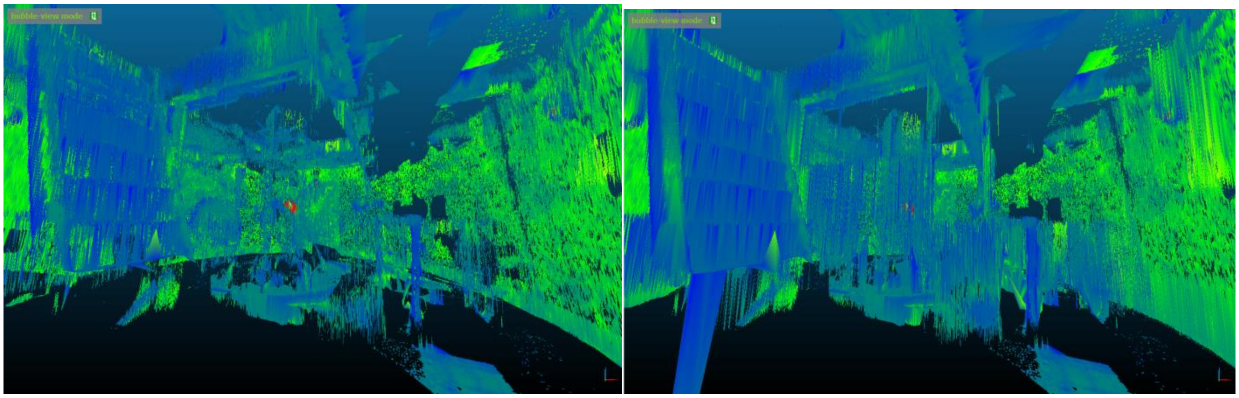


Figure 52: 2nd floor subsampled mesh 0.2m and 0.5m mesh

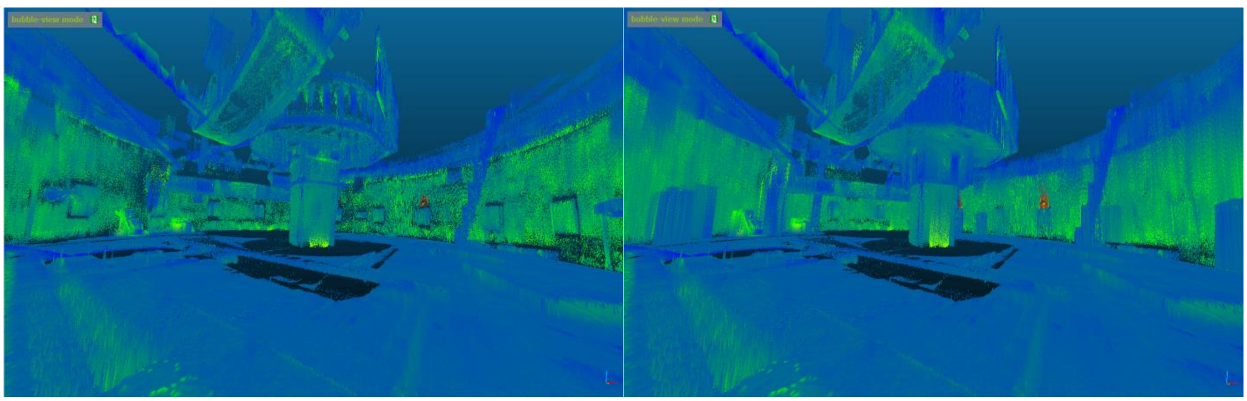


Figure 53: 1st floor part 1 original mesh 0.2m and 0.5m mesh

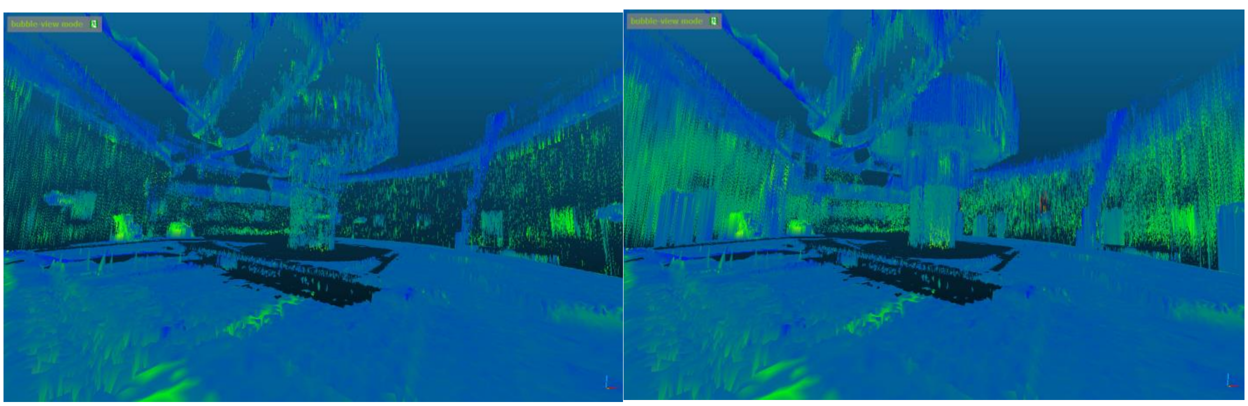


Figure 54: 1st floor part 1 subsampled 0.2m and 0.5m mesh

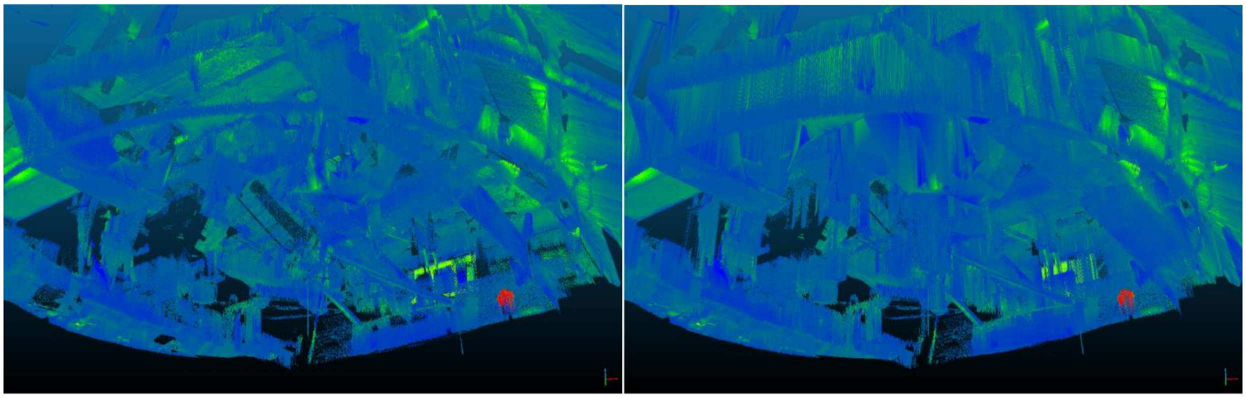


Figure 55: 1st floor part 2 original 0.2m and 0.5m mesh

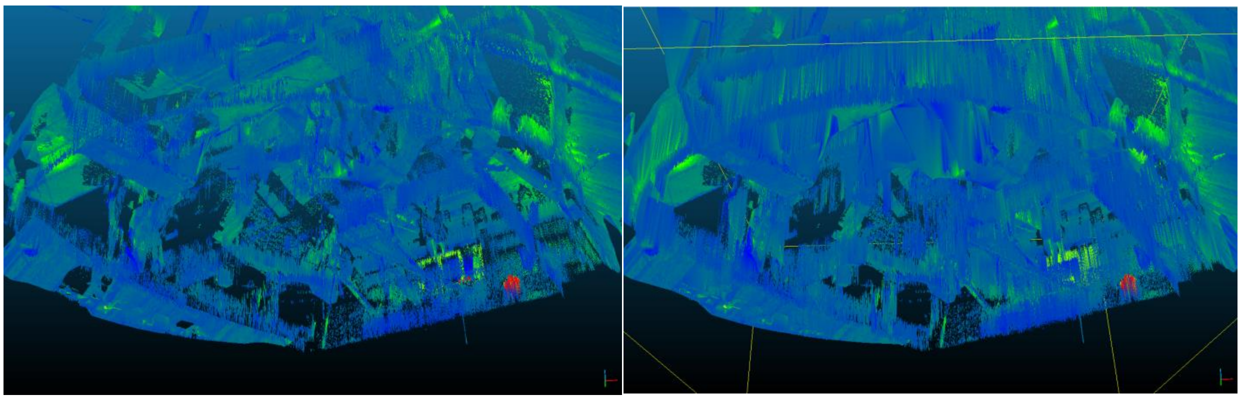
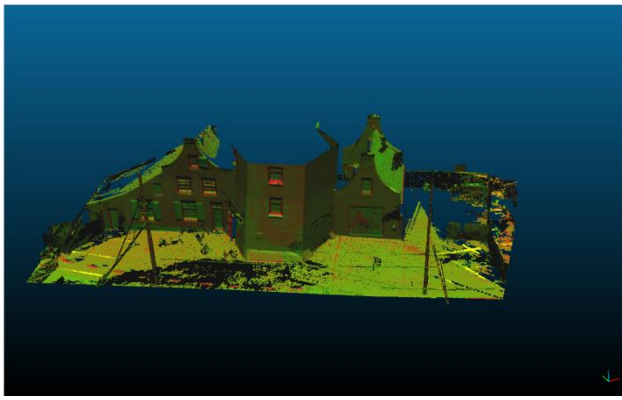


Figure 56: 1st floor part 2 subsampled 0.2m and 0.5m mesh

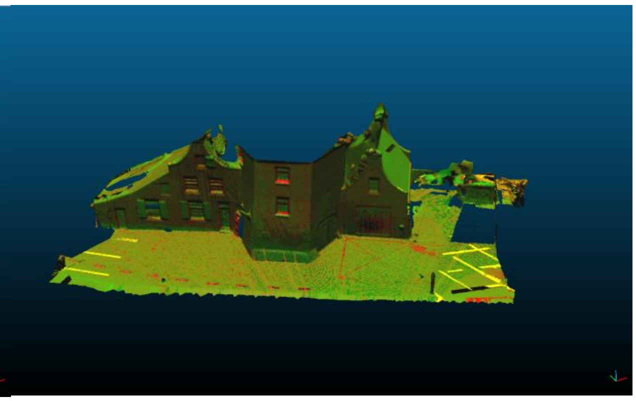
Appendix II

Below, the remaining results are shown in a series of figures regarding the ball pivoting and Poisson reconstruction algorithm for the balcony, the exterior excluding the balcony, the remaining floors and the roof and blades.

<u>Exterior below balcony</u>



a



b

Figure 10: ball pivoting and Poisson method

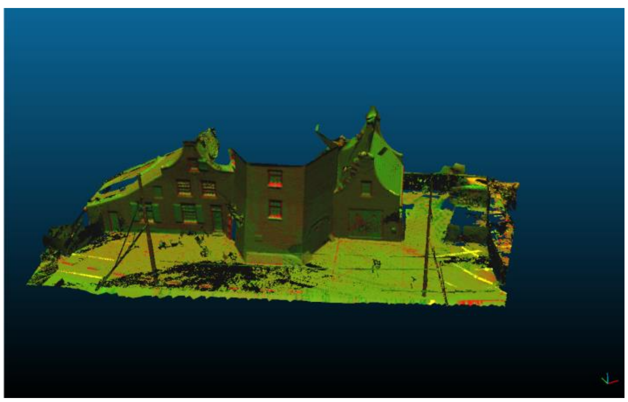
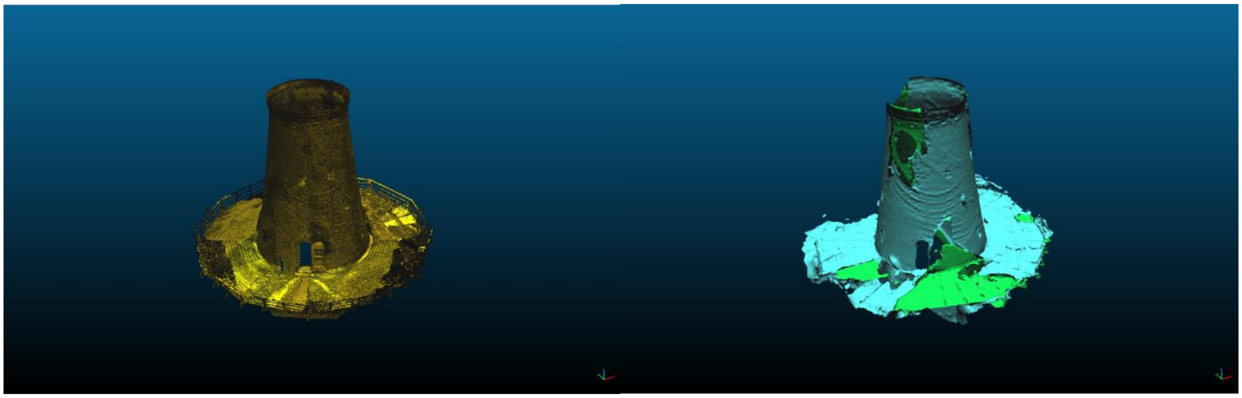


Figure 11: below balcony merged

Balcony



a b

Figure 59: ball pivoting and Poisson method

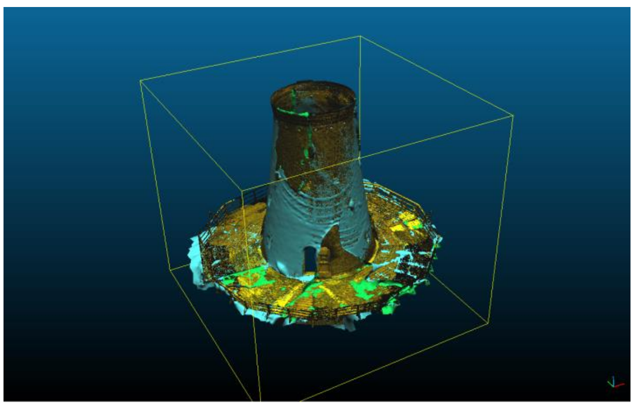
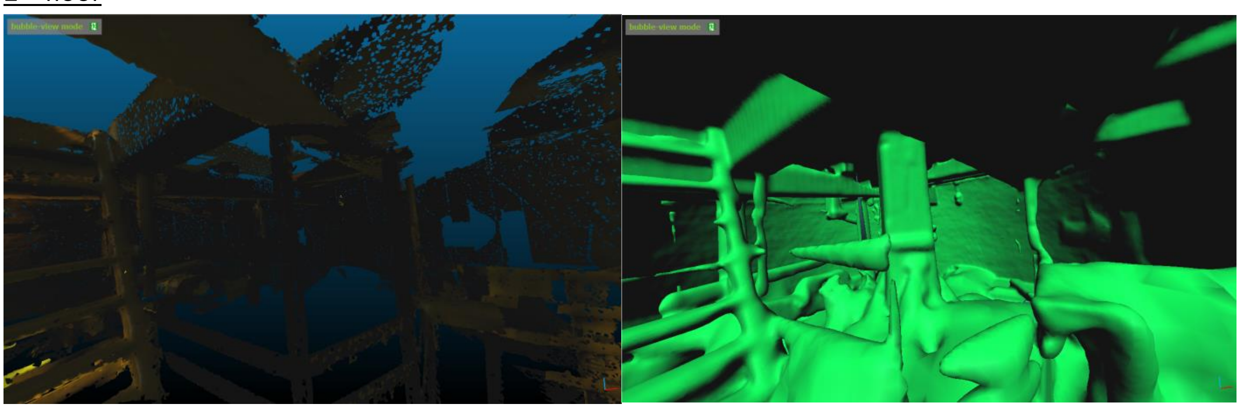


Figure 60: balcony both opened at same time

2nd floor

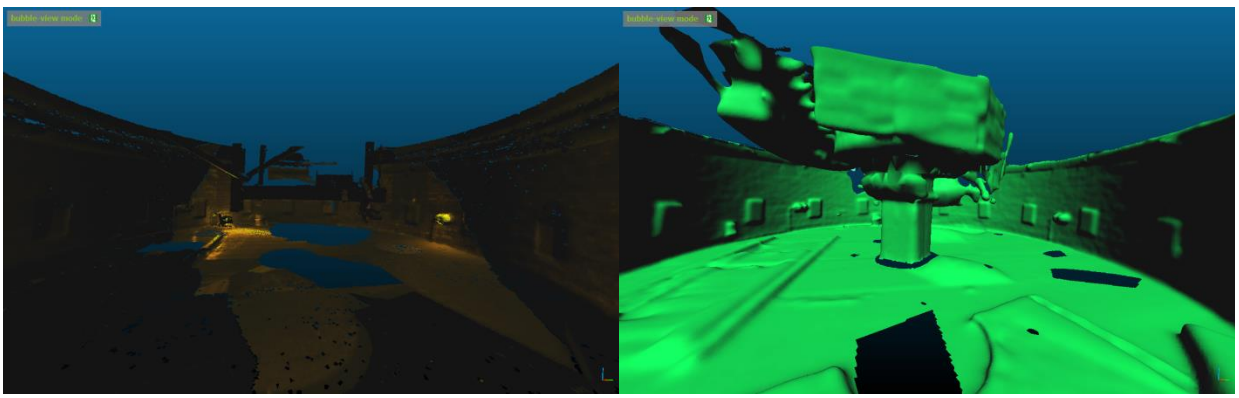


a b

Figure 61: ball pivoting and Poisson method

Figure 62: second floor merged

1st floor part 1



a b

Figure 63: ball pivoting and Poisson method

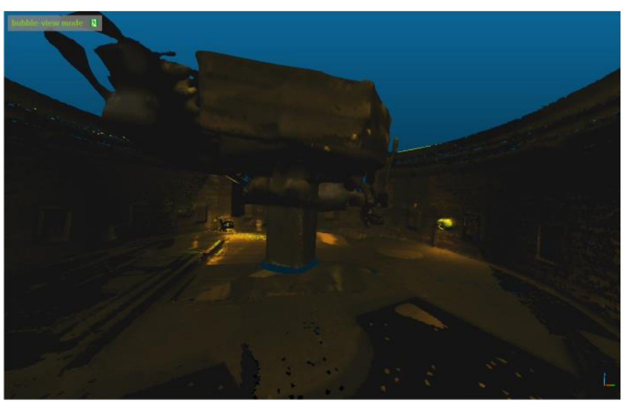
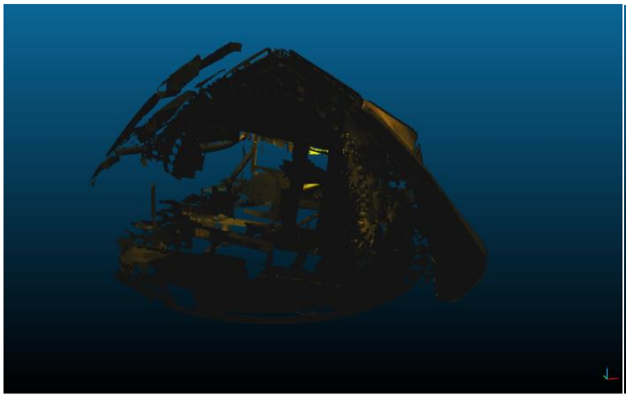
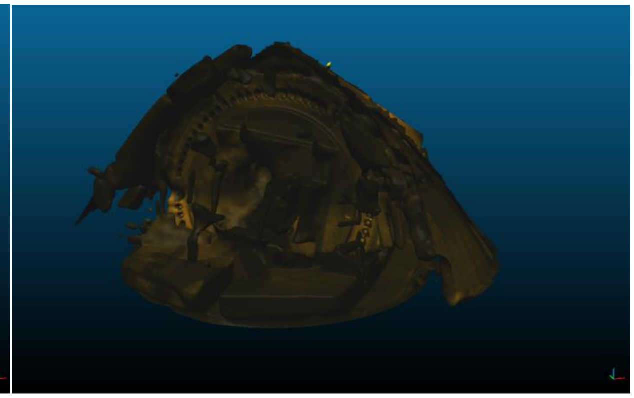


Figure 64: 1st floor part 1 merged total

1^{st} floor part 2





a b

Figure 65: Ball pivoting and Poisson method

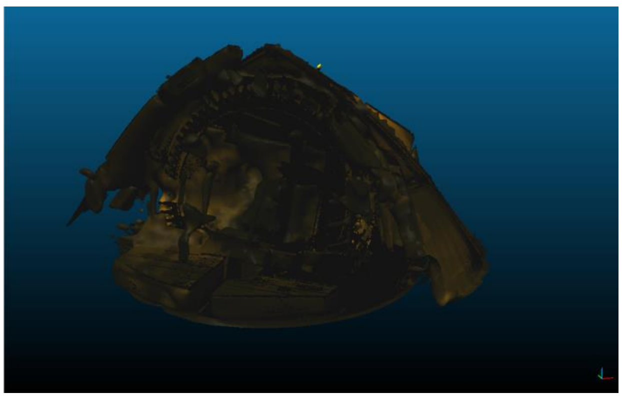


Figure 66: 1st floor part 2 total merged