Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Fachbereich Landschaftswissenschaften und Geomatik

Studiengang Geodäsie und Geoinformatik

Improved Meshing Pipeline for 3D-data Generated from Terrestrial Laserscans

Master thesis

prepared by: *Oleksandr Bocharov*

for obtaining the academic degree
"Master of Engineering" (M.Eng.)

Examiner: Prof. Dr.-Ing. Sven Braemer
Second Examiner: Prof. Anatolii Tserklevych
Submitted on: 24.07.2017
URN: **urn: nbn:de:gbv:519-thesis 2018 - 0041 - 9**

Declaration for the Master’s Thesis

I confirm that this Master's thesis is my own work and I have documented all sources and material used. This thesis was not previously presented to another examination board and has not been published.

Place and date

Signature

ANNOTATION

Oleksandr Bocharov

Improved Meshing Pipeline for 3D-data Generated from Terrestrial Laserscans

Studiengang Geodäsie und Geoinformatik

Hochschule Neubrandenburg

Lviv 2018

The task of this Master thesis is to identify a practicable pipeline to transfer data from 3D-Laserscans into the mesh modeling software ‘Blender’.

The master thesis includes the following sections:

1. Abstract;
2. Introduction;
3. 3D terrestrial laser scanning;
4. Laser Scanner Faro Focus 3D;
5. Realisation of laser scanning;
6. Processing laser scan data in Faro Scene
7. Meshing;
8. Data transfer from Faro Scene to Blender;
9. Conclusions;
10. References.

The result of this work identifies the most practicable way and the currently most suited file format for processing 3D laser scanner data in Faro Scene with further transferring to Blender.

The master thesis contains 6 tables and 33 figures.

Keywords: 3D laser scanning, mesh, point cloud, file format.

ABSTRACT

This master thesis shows problem of transferring data to mesh modeling software. The data source was a 3D laser scanning of church. This laser scanning was done in 2013 with Faro laser scanner.

All steps for processing laser scanner data were executed in Faro Scene. Than a mesh was created. A File with 3 test scans was converted in all file formats which are acceptable for importing in Blender software.

In final part of master thesis the results are described of importing data in Blender. Each file format is estimated with relevant parameters to choose the most suitable and practicable format for data transfer realization.

CONTENT

Introduction 6

1. 3d Terrestrial laser scanning 7
   - 1.1. Sphere of usage 7
   - 1.2. Types of laser scanners 9
     - 1.2.1. Triangular laser scans 10
     - 1.2.2. Pulse based (Time-of-flight) Laser Scanners.......................... 11
     - 1.2.3. Phase Based Scanners 12
2. Laser Scanner Faro Focus 3D 14
3. Realisation of laser scanning 18
   - 3.1. Object of scanning... 18
   - 3.2. Scanning technology 20
4. Processing laser scan data in Faro Scene... 23
   - 4.1. Faro Scene software 23
   - 4.2. Registration of scans.. 24
   - 4.3. Cleaning the noise and creating point cloud............................... 27
5. Meshing 30
   - 5.1. Mesh... 30
   - 5.2. Creating a mesh 33
6. Data transfer from Faro Scene to Blender 36
   - 6.1. Blender Software.. 36
   - 6.2. Types of used files 37
     - 6.2.1. Stanford file format 38
     - 6.2.2. STL file format 40
     - 6.2.3. Virtual Reality Modeling Language file format 42
     - 6.2.4. Wavefront file format 44
     - 6.2.5. AutoCAD file format... 44
     - 6.2.6. XYZ file format... 46
7. Conclusions 47
8. References.. 51

INTRODUCTION

In the last years the need for 3D technologies increased rapidly. At the same time the ability of 3D technic has grown as well. Now it is widely used not only in geodesy and architecture sphere, but in digital entertainment such as video games and 3D cartoons, virtual excursions to museums and historical sites.

These spheres demand higher resolution to improve the quality of their products. They demand high resolution data. This causes some problems like storage of data and processing, or representation. The usual point cloud is not the best practicable way for all this type of manipulation. In this case the method of meshing 3D point clouds can be helpful.

Intelligent meshing reduces loads on the PC and the size of processed files. It also allows simplifying of created models. For example the scanned wall contains a lot of points, but it may for certain purposes be sufficient to present this wall with four points in corners and one face. So transferring raw laser scan data and point clouds to mesh modeling software is an important task, and it needs to be done in a way that does not loose the high-quality and realistic simulation of scanned objects.

1. 3D TERRESTRIAL LASER SCANNING

1.1. Sphere of usage

3D terrestrial laser scanning is a systematic process of determining points coordinates. The results of laser scanning are used to create wireframe, surface or volume models with the help of CAD-systems. 3D scanners is the name of devices, which are used for laser scanning. These devices not only simplify the process of creating 3D models, but also solve this task with appropriate accuracy and reliability.

The main advantages of laser scanning:

- A significant reduction in the terms of time and work
- Accuracy, as in traditional geodetic methods
- Easy integration into workflows
- Compatibility with traditional geodetic methods
- Receipt of high density information about the object
- Ability to reuse the results of measuring
- Getting precise and visible results already at the preliminary stage of work

Further Advantages of the 3D scan method are fast three-dimensional visualization of objects, high precision, portability, convenience, a wide range of objects, accurate color transfer. Because of these features 3D scanners can be used for a wide range of tasks in industries, science, medicine and arts. In particular, they successfully solve the problems of reverse engineering, control of the form of objects, the preservation of cultural heritage, used in museum affairs, medicine and design. Therefore, they are nowadays indispensable in all cases when it is necessary to register the shape of the object with high accuracy and in a short time. Three-dimensional scanners allow to simplify and improve manual work, and sometimes even perform tasks that with other methods are impossible.

Laser scanning is used in the following areas:

- Fast and accurate measuring at all stages of construction works
- Monitoring of deformations for existing buildings and objects under construction
- 3D cadaster

- Process and store such information as separate elements of decoration, moldings, any other fragments of historical buildings or monuments, the entire archaeological excavation area or their separate fragments
- Precise definition of the ground volume which should be moved
- Work with remote control in places dangerous to people

1.2. Types of laser scanners

Scanning is conducted for different purposes on different objects. Different types of scanners are manufactured to provide specialized performance, which are categorized according to certain features (Fig. 1.1).

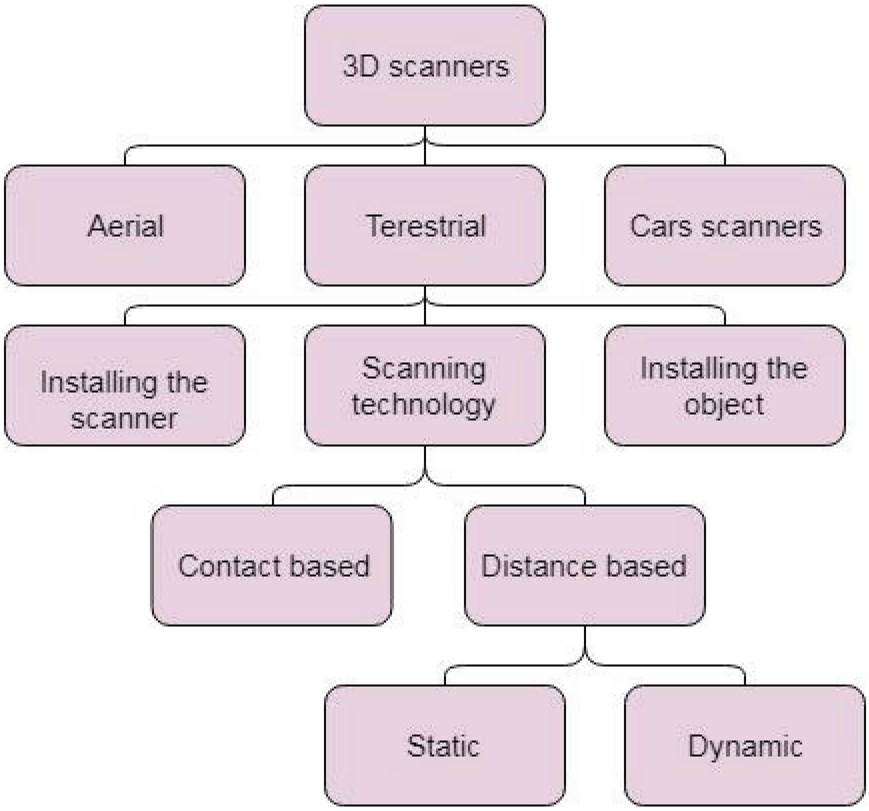

Figure 1.1: Types of Laser Scanner

Aerial scanners are installed on the aircraft and scan the large area of earths surface. One of the key points of the work of these scanners is the presence of a special positioning system IMU (Internal Measurement Unit). It allows to trace the placement scanner, taking into account speed and direction of flight, and spatial orientation of the plane. These scanners do not give high-precision data in the sub-millimeter area, but they are able to scan large areas in a short period. Car laser scanners combine features of aerial and terrestrial laser scanners. Data collection in dynamic mode using inertial and satellite navigation systems are typically common features to aerial and cars scanners.

The way to install the scanner depends on the dimensions the device itself. Based on this they can be divided into stationary and portable.[1]

1.2.1. Triangular laser scans

In 2007 at the International Exhibition and Scientific Congress “GeoSiberia 2007”, (Norbert Pfeifer and Christian Briese, 2007) explained the principles of triangular laser scans. They say that obtaining even higher accuracy with laser scanning, range may not be determined directly, but by angle measurements. In a triangulating laser scanner(Fig. 1.2), the laser ray is expands in order to form a plane, rather than a beam. With the help of a rotating mirror, this plane is passed through the object space. For one position of the plane, one angle of the mirror, the intersection of this plane with the surfaces in object space results in one band, or many bands. The object space is, at the same time, projected by a lens on an image plane. The image covers the entire scene, but only the band is of interest. It can be extracted by calculating the difference image of two images. The first image is the scene without the laser line. The second image contains the “laser light” bands. The curves in the image plane form a beam of rays, connecting the plan of the bands with the projection centre. The intersection of this bundle with the plane of laser energy determines the position of the points in object space. This method of scanning is limited in depth, because the precision of the intersection reduces with extend. The basis distance from the transmit to the camera, can not be made too big because of practical reasons. This type of scanners is limited to ranges of one or a few meters. The accuracy is normally better than ±1mm. The method described measures not one point at a time, but a order of points along the “laser light” bands on the purpose objects. After a few seconds the entire can be scanned. The amount of points depends, among other factors, on the resolution of the camera. [2]

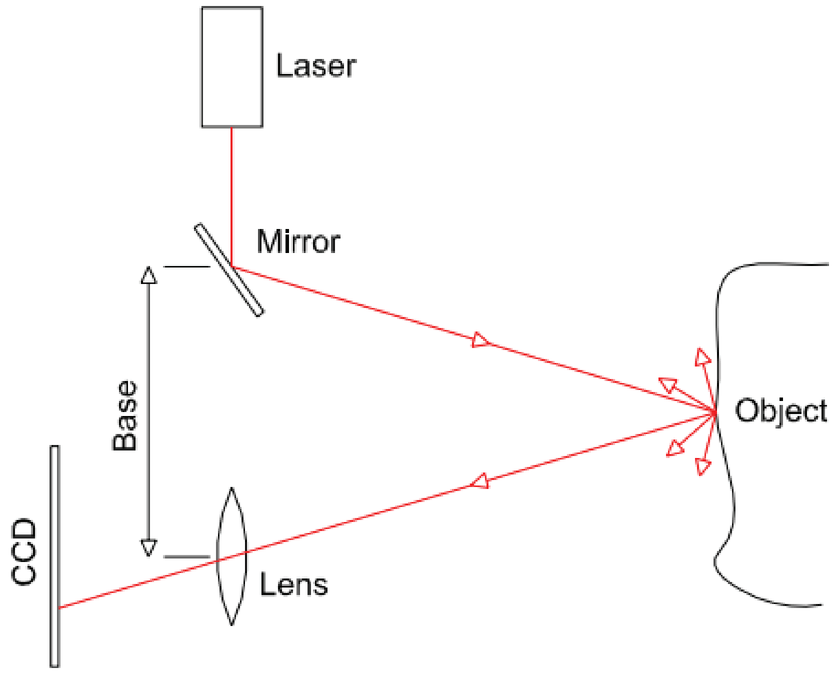

Figure 1.2: Triangulation Principle in Laser Scanning.[3]

1.2.2. Pulse based (Time-of-flight) Laser Scanners

Pulse based method is explained in training materials of project “Learning tools for advanced three-dimensional surveying in risk awareness project”, (Boehler, W., 2001). Light waves spread with a finite and constant velocity in a certain environment. When the time retention created by light spreading from an transmitter to aim and back to the emitter can be defined, the distance to that aim can be estimated using the formula(1.1):

$$D = \frac{(c*t)}{2} \qquad (1.1)$$

with c = speed of light in air and t = time between sending and receiving the signal

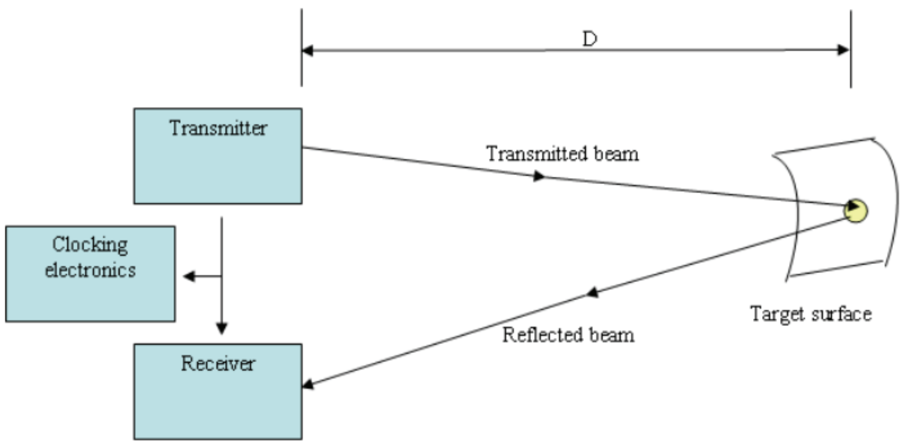

Figure 1.3: Time-Of-Flight Laser Scanner Principle

The current approved value for the speed of light in a vacuum is c = 299,792,458 m/s. In case of spreading the light waves in air a correction factor equal to the refraction index must be applied to c. Taking into account c, one can calculate it takes 3.33 nanoseconds to pass 1 meter. To attein a point precision of 1 mm, time retention 3.33 picoseconds needs to be able to be measured.

Pulsed (time-of-flight) scanners do not use uninterrupted laser rays, but make use of laser impulses. They scan their views zone one point at a time by changing the direction of range-finder. The view direction of the laser range-finder is changed by a deflection parameter. Usually pulsed 3D laser scanners can measure nearly 2,000~50,000 points every second.[4]

1.2.3. Phase Based Scanners

Principles of phase based scanners are described in “Practical application of 3D laser scanning techniques to underground projects”. Compared to the puls based scanner, phase based type of scanner has a high speed scanning rate and better accuracy, but a short distance in the range of tens of meters. In this case, the transmitted beam is modulated by a harmonic wave and the distance is calculated using the phase difference between the transmitted and received wave. The Phase-based scanner has a higher precision, in the domain of millimetres, and higher measurement rates up to one million points per second, can be obtained applying the phase shift measurement principle. A continuous wave laser is used as the carrier for a signal modulated onto it, typically using amplitude modulation. The phase of the emitted and the received signal are compared. The measured distance in one direction is:

$$\gamma = \Delta\varphi/(2^*\pi)^*\lambda/2 + \lambda/2^*n \qquad (1.2)$$

where $\Delta\varphi$ is the relation between phase differences, λ is the wavelength in metres, and n is the unknown number of full wavelengths between the transmitter and the aim surface(Fig. 1.4) Selected λ value means that there is a unique measurement range of λ/2. All measurements to objects further away will be embedded into the first λ/2 interval. The accuracy of the measurement is one percent of the phase and can be improved. The improvement can be done via using more than one modulation wavelength. The long wavelength defines the unicity range and the shortest wavelength defines the accuracy that can be reached.[5]

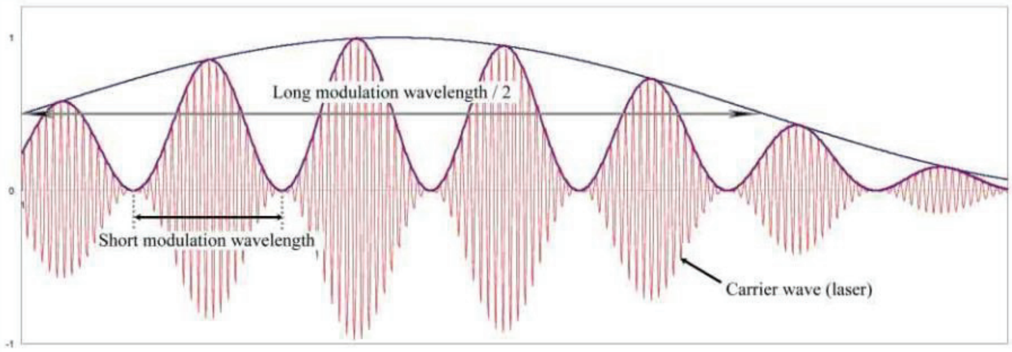

Figure 1.4 Schematic Drawing of Two Modulation Wavelength And Carrier Wave For Phase-Based Laser Ranging

2. LASER SCANNER FARO FOCUS 3D

Faro Laser Scanner Focus 3D (Fig. 2.1) “is a high-speed three-dimensional laser scanner for detailed measurement and documentation. The Focus3D uses laser technology to produce detailed three-dimensional images of complex environments and geometries in only a few minutes. The resulting images are an assembly of millions of 3D measurement points.” [6]

Figure 2.1: Faro Laser Scanner Focus 3D

In principle, the Focus 3D works by spreading an infrared laser ray into the center of its rotative mirror. The mirror changes the laser ray on a vertical rotation into the scanned object. The light from objects is then reflected back into the scanner (Fig. 2.2).

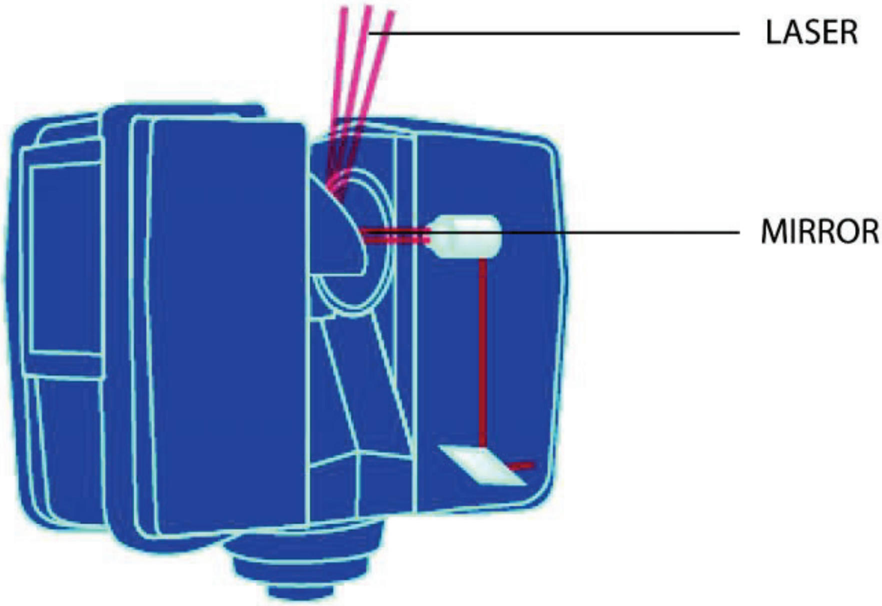

Figure 2.2 Laser Deflection

To measure the distance, Focus 3D uses phase shift method, where permanent waves of infrared light of changing length are projected out from the scanner. In case of contact with an object, they are reflected. The ratio incoming light / reflected light on the object is called ‘albedo’. A certain amount of the reflected light is reflected back to the scanner. This intensity is called ‘reflectance’. The distance from the scanner to the object is determined by measuring the phase shifts in the waves of the infrared light. Hypermodulation enlarges the signal-to-noise ratio of the modulated signal with the help of a special modulation method. “The x, y, z coordinates of each point are evaluated by using angle encoders to measure the mirror rotation and the horizontal rotation of the Focus 3D. These angles are encoded at the same time with the distance measurement. Distance, vertical angle and horizontal angle make up a polar coordinate (δ, α, β), which is then transformed to a Cartesian coordinate (x, y, z). The scanner covers a 360° x 305° field of view"[6] (Fig. 2.3).

Moreover, the Focus 3D identifies the reflectivity of the scanned surfaces by measuring the intensity of the received laser ray. At all, bright surfaces reflect a larger amount of the emitted light than do dark surfaces. This reflectivity value is used to appoint a corresponding grey value to each single point.

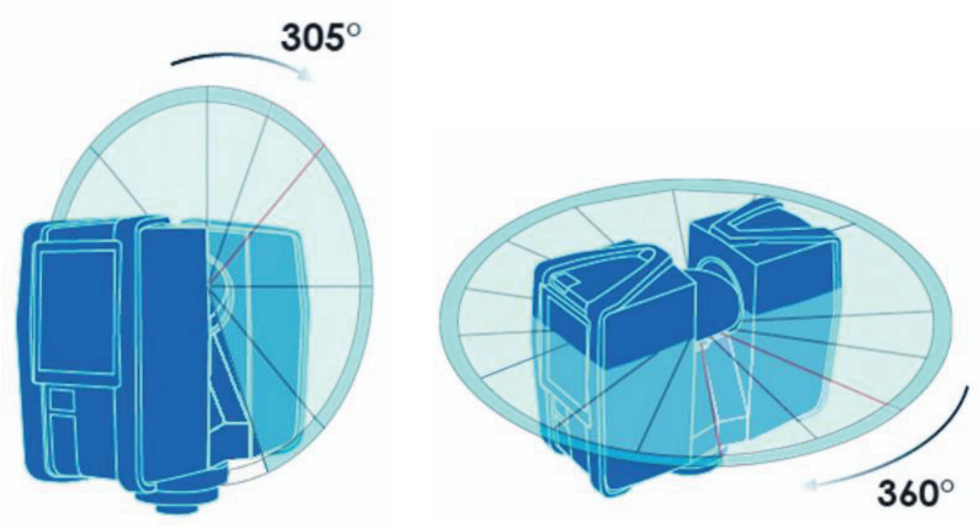

Figure 2.3: Vertical and Horizontal Rotation

The measurements of one point are repeated up to 976,000 times per second. The result of measurement is “a Point Cloud, a three-dimensional dataset of the scanner's environment (hereinafter referred to as the „laser scan“ or „scan“). Depending on the selected resolution (points acquired per rotation) each point cloud consists of millions of scan points".[6]

The laser scans are recorded to the changeable SD memory card, that allows secure transfer to SCENE, FARO’s point cloud manipulation software.[6]

Table 2.1

Specifications of Faro Focus 3D

Ranging unit	
Unambiguity interval:	153.49m (503.58ft)
Range Focus3D S 120 :	0,6m - 120m indoor or outdoor with low ambient light and normal incidence to a 90% reflective surface
Range Focus3D S 20:	0,6m - 20m at normal incidence on >10% matte reflective surface
Measurement speed (Pts/Sec):	122,000 / 244,000 / 488,000 / 976,000
Ranging error :	±2mm at 10m and 25m, each at 90% and 10% reflectivity
Colour unit	
Resolution:	Up to 70 megapixel colour
Dynamic colour feature:	Automatic adaption of brightness
Deflection unit	
Field of view (vertical/horizontal):	300° / 360°
Step size (vertical/horizontal):	0,009° (40,960 3D-Pixel on 360°) / 0,009° (40,960 3D-Pixel on 360°)
Max. vertical scan speed:	5,820rpm or 97Hz
Laser (Optical transmitter)	

Laser power (cw Ø):	20mW (Laser class 3R)
Wavelength:	905nm
Beam divergence:	Typical 0.19mrad (0.011°)
Beam diameter at exit:	Typical 3.0mm, circular
Data handling and control	
Data storage:	SD, SDHC™, SDXC™; 32GB card included
Scanner control:	Via touchscreen display and WiFi
New WiFi(WLAN) access:	Remote control, Scan Visualisation and download are possible on mobile devices with Flash®
Multi-Sensor	
Dual axis compensator:	Levels each scan: Accuracy 0,015°; Range ± 5°
Height sensor:	Via an electronic barometer the height relative to a fixed point can be detected and added to a scan.
Compass:	The electronic compass gives the scan an orientation. A calibration feature is included.
Power supply voltage:	19V (external supply), 14.4V (internal battery)
Power consumption:	40W and 80W (while battery charges)
Maintenance / calibration:	Annual
Parallax-free:	Yes

3. REALISATION OF LASER SCANNING

3.1. Object of scanning

The wooden church St. Archangel Michael in Komarno (Fig. 3.1) it is a national architectural monument. Komarno is a city of Gorodotsky district of Lviv region. It is located in the southwestern part of the Lviv region on the banks of the river Vereshchitsa (tributaries of the Dniester River). The distance to the district center is 20 km, to Lviv is 45 km, to the railway station 4 km. The city owns 3,778 hectares of land, including the city's territory of 732 hectares.

Figure 3.1: The Wooden Church St. Archangel Michael

The size of the church is 16,5 × 7,2m and it is covered by three domes. The inscription 1754 is located on the wall of the church and refers to the restoration. The memorial actually is older. During the restoration in 1891 the western side, a dome, a roof, and a floor were restored. The bell tower (Fig. 3.2) was restored in the same year. In the 1920’s, a northern sacristy was dismantled. The restoration of 1965-1967 returned the church to its original appearance. The wooden bell tower is located about 20 meters west of the church of 5,2 × 6,0 m column structure. The bell tower was restored in the same year. Once there were three big bells in the bell tower, but the Austrians melted them into cannons for the Austrian authorities. Almost the same story happened with the bells that were cast later in 1931 in the city of Lviv by the Brilin brothers. All the people of Komarna gave old coins to make the bells. And again, three bells were cast and were given with the name Michael, Vladimir and Ivan. In the same

1931, they were consecrated and there was inscriptions on each bell: “Foundation of the Holy Trinity Fellowship in Komarno”.

Figure 3.2: The Bell Tower

The old wooden church until 1810 was mainly for people who lived in the suburbs of Komarna and on the side of Shchyrets. When there was a big fire in Komarno and many buildings and the parish church of St. the Apostles Peter and Paul, which was situated in the center of the city burned down. After the fire in 1810, the old wooden church of the Holy Archangel Michael became the main Parish Church. There was not any church in the center of the city for 39 years and only in 1849 was built a new church.[7]

3.2. Scanning technology

St. Archangels Michaels church and its bell tower were scanned in 2012. For realization of this scanning 22 exterior and 11 interior scans were completed (Fig. 3.3).

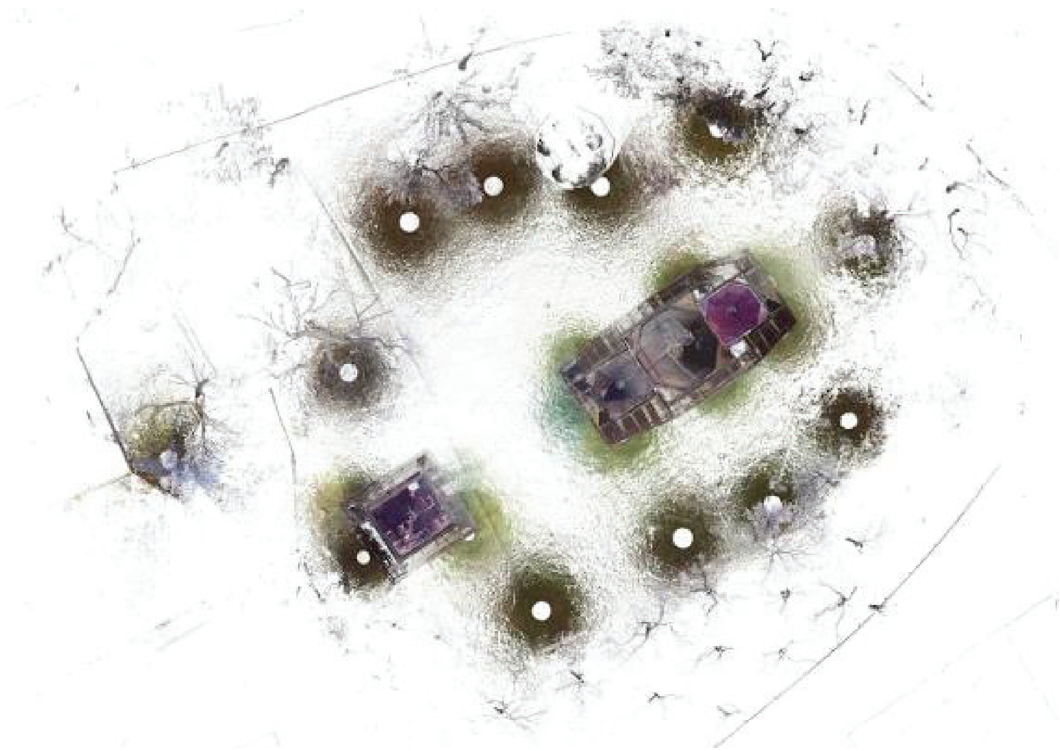

Figure 3.3: Scheme of Laser Scanner Stations

But in this master thesis only three exterior scans were used, because of high loading on the RAM. Two of them are far range scans and one is close to the object (Fig. 3.4, 3.5).

Figure 3.4 Panoramic Image from Far Range Station

Figure 3.5 Panoramic Image from Far Range Station

The Technology of the realization of the laser scanning depends on the geometry of the object. Sometimes scanning needs a lot of stations to scan all details. The reason for this is the existence of ‘blind zones’. In case of St. Archangel Michael church it was very important to make scans under the roof visor around, because it gives the complete geometry information. In addition, close position to the object gives better representation of the object surface texture (Fig. 3.6)

Figure 3.6 Panoramic Image from Close Range Station

There often is a need to bring all scans in one coordinate system (a process called ‘registration’), because each individual scan uses its own coordinate system. To do this, during scanning on or near the object, special registration targets (Fig. 3.7) are used to adjust the point clouds obtained from different points of scanning. For adjusting of clouds, at least three marks are required for each point of installation of the scanner. These three points with marks should be visible from adjacent points.

Figure 3.7: Spherical Registration Targets

There are also a lot of other forms of registration targets, but the spherical form has an advantage because it has the same external shape from any directions. Usually buildings do not contain inside and outside of rounded objects, therefore, during processing, there is not a high probability to confuse other objects with spheres. In case of this confusing manual adjustment solves this problem.

Figure 3.8 Configuration of Targets

4. PROCESSING LASER SCAN DATA IN FARO SCENE

4.1. Faro Scene software

The next step after scanning stage is transferring data into 3D visualization software. Scene was chosen for this purpose.

Scene software is specifically designed for all Focus, Freestyle and third-party laser scanners. It supports processing and managing scan data by using real time, on-site registration, automatic object recognition, scan registration, and positioning. It generates high-quality data in RGB color quickly and conveniently by incorporating images from automated targetless and target-based scan positioning.

Once SCENE has processed the scan data users can perform simple measurements, create 3D visualizations or export to various point cloud and CAD formats. In addition SCENE has a feature called Virtual Reality View, allowing users to experience and evaluate captured data in 3D. This is essentially an octree based 3D rendering pipeline.[8]

FARO has its own data format type *.fls. A typical *.fls file contains:

A matrix of distances obtained from time of flight/phase shift, arranged in row and column according to the sequence of the scanning process.

In addition a matrix with gray values for each scanned point (reflectance)

In addition metadata with time, GPS, Altitude, Compass, correction values etc.

In addition RGB-pictures of the scene, taken from different angles of rotation for each scanner position

On all this data a lossless compression algorith is applied to effectively store the data in the manufacturer specific format *.fls.

After those *.fls files are imported in Scene software, registration of scans should be implemented.

4.2. Registration of scans

The following three methods of registration are possible in FARO Scene:

- Automatic registration
- Manual registration
- Visual registration

Taking into account that only 3 scans were used, the optimal method of registration was ‘manual’. The sphere fitting algorithm is used to detect tie points and to transform the different point clouds into a common reference system is described in (Bienert, Maas, 2009).[9]. As described above, the used tie points were white spheres. The registration process is done by marking all tie points in Faro Scene in each 2D intensity image (Fig. 4.1)

Figure 4.1: Marking spheres in images

Corresponding spheres are identified by the user itself and obtain identical names. Two options are given for calculating the centre point of each sphere:

- first the sphere shall fitt in the range data with a fixed radius and
- second a sphere shall fit by adjusting the radius as well.

Afterwards a transformation with the fitted centre points is done.

Research (J. L. Lerma and D. García-San-Miguel, 2014) explains the mathematical realization of geometrical calibration. Geometric calibration based on point features demands setting a reference framework. This network determines the object space coordinate system. “This system will be a three-dimensional (3D) coordinate system defined marginally by the target centroids. The measurements of the same targets by the terrestrial laser scanner will be referenced to the laser scanner co-ordinate system; origin and axes are defined by the instrument itself (the origin of the range measurements fits theoretically the intersection of three axes: collimation axis, trunnion axis and vertical axis). Both co-ordinate systems will be related by a 3D rigid-body transformation with three rotations and three translations that constitute the six exterior orientation parameters (EOP) for each scan station.[10] The EOP it is a connection between both co-ordinate systems. "Three observation equations (Eq. 4.1) are developed for each reference point measured in both co-ordinate systems.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = R_\kappa R_\varphi R_\omega \begin{Bmatrix} X_i - X_s \\ Y_i - Y_s \\ Z_i - Z_s \end{Bmatrix} \qquad (4.1)$$

Where x_i,y_i, z_i are the co-ordinates of object point '*i*' in the scanner system; R_κ, R_φ and R_ω are the matrices for the three rotations about the X-, Y- and Z-axes, respectively; X_i, Y_i, Z_i are the object space co-ordinates of object point '*i*'; X_s, Y_s, Z_s are the object-space co-ordinates of the laser scanner's origin. The translation between the co-ordinate systems will be given by (X_s, Y_s,Z_s)." [10]

Registration was realized with the following results (Tab. 4.1, 4.2, 4.3): maximal point error = 10.0 mm, average point error = 8.3mm, minimal overlapping = 40.7%. The results are with tolerance deviation.

Table 4.1

Scans errors

Cluster/Scan	Verbindungen	Max. Punktfehler [mm]	Mittlerer Punktfehler [mm]	Min. Überlappung
Dyplom058	2	7.6	7.4	59.8 %
Dyplom059	2	10.0	8.6	40.7 %
Dyplom057	2	10.0	8.8	40.7 %

Table 4.2

Errors details

Cluster/Scan 1	Cluster/Scan 2	Punktfehler [mm]	Überlappung
Dyplom058	Dyplom059	7.2	59.8 %
Dyplom057	Dyplom058	7.6	62.0 %
Dyplom057	Dyplom059	10.0	40.7 %

Table 4.3

Differences of inclinometer

Cluster/Scan	Scan	Diskrepanz [deg]
Dyplom057	Dyplom057	0.0341
Dyplom058	Dyplom058	0.0354
Dyplom059	Dyplom059	0.0106

After that from the originally row and column sorted scans a so called 'scans point cloud' can be created (Fig. 4.2).

In point clouds the data are differently organised than in Row/Column. This is done in order to allow movement in the 3D world. The hugh amount of data can not be completely transformed to the camera pose and rendered 45 times per second (frame rate). Some of the data must be omitted to achive anacceptable frame rate.

For that pupose data is organised in a way so that for a known camera pose it becomes very fast obvious, what needs NOT be rendered, because it is invisible from that point of view. An octree is a structure that does exactly that. The 3D Space is divided into 8 (oct) disjunct quaders, which are themselvs subdivided into 8 and so on until screen resolution of a pixel.

Given camera pose and field of vision of the 3D-view now the quaders to be rendered may be determined very fast.

Figure 4.2 Scans Point Cloud in Scene Software

A digital camera is included in FARO laser scanner and it takes photos after scanning. They can later be used to colorize the scan. For this purpose the matrix with gray-values (reflectance) gets a sister matrix. It contains for each scanned point RGB-values instead of gray values. During colouring an algorithm calculates from the coordinates in row and column the corresponding point in one of the RGB-images that were taken after the scanning (by projection). This RGB-value is then inserted into the sister matrix. If a scanned point does not correspond to exactly one pixel in an image, usually a bilineary interpolation is performed.

4.3. Cleaning the noise and creating point cloud

When laser scanning is implemented inside of some historical object all scanned points are used for creating model or point cloud. But in this case outside scans were used and object is not close around scanning station, so not all scanned points belong to the building. Unnecessary points (noise, trees, people, grass...) can be deleted.

So the next task is to clean up noise and points which are not of any interest. There are the several ways to do this:

- Cleaning raw scans
- Cleaning scans point cloud
- Cleaning project point cloud

The disadvantage of cleaning row scans is that in case of noise location on the scans intersections, the points from the same source of noise have to be deleted from each scan. This not practicable and the operator has to do the same work several times. The most rational way is to clean it in the scan point cloud. If it later turns out to be necessary, the deleted points can be recreated from the raw scans. Cleaning scans point cloud has the advantage in avoiding double cleaning and after creating project point cloud it stays the same after updating or recreating. When a project point cloud is cleaned up and updated or recreated all the noise from not cleaned scans cloud will be brought in.

The noise means scanned parts of equipment or silhouettes of people who were doing laser scanning. It can be a 100% complete object which shall not be shown, or just some points which were scanned during moving of noise source. So the territory of the bell tower was deleted. The mirror surfaces are not presented in point cloud. As the church has windows and their glass material has the ability to reflect, noise around the window appears (Fig. 4.3).

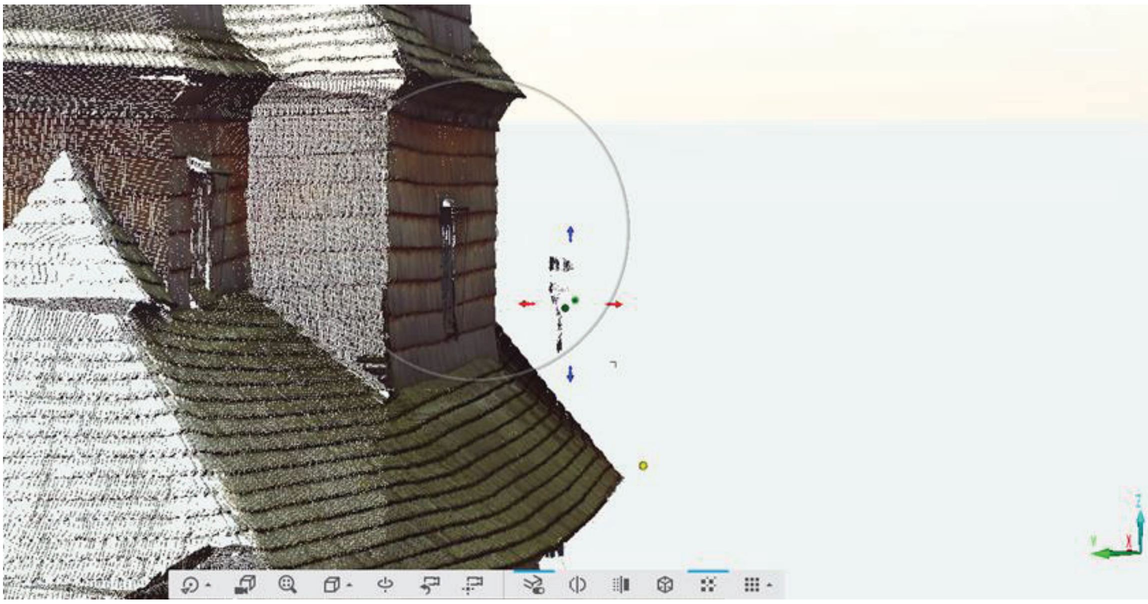

Figure 4.3: Glass reflection noise

“Scanning reflective surfaces without further precautions yield incorrect 3D point clouds, since the emitted laser light is reflected by the surface and a wrong distance is determined.” [11] The windows are presented, because glass reflects rays only at a certain angle. The action of cleaning up was done with Clipping Box function. This function allows to create the area which can be scaled and rotated to separate the objects which have to be processed.

Then point cloud should be created. Point cloud it is array of points with defined coordinates X, Y, Z in scanners coordinate system connected together. Scene project contains all projects scans and keeps it in *.lsproj file format. Point cloud of three scans has 20 591 723 points. Storing laser scanning data as point cloud is the most practicable and popular. Every point has not only the information about position in three-dimensionally coordinate system but also about color. Information about the color is represented as three values from RGB channel of laser scans camera. These three values are shades of red, green and blue color and theirs value can range between 0 and 255. So each scanned point is described with six values.

Figure 4.4: Point cloud of St. Archangel Michael church

5. MESHING

The data can be transferred from Faro Scene as single scan point cloud, point cloud or mesh. For the last option FARO has implemented a meshing algorithm. Meshing of scanned points is not the core task of a laser scanner. So, while most software packages offer this option, the implementations are usually limited in capability and the user has few chances to influence parameters of the mesh creation. There is more powerfull software on the market, which is specialized to perform this job. Especially the mesh -modelers like 3DS - max, Maya or Blender have advanced meshing algorithms. It is the task of this thesis to investigate the point, up to which the laser scan software still performs well and then to transfer the data into a mesh modeller which can cope better with subsequent tasks (non-manifold checking, watertight, smoothing, uv-unwrapping, etc.). This transition point is to be found before or after the creation of a mesh.

5.1. Mesh

A mesh is "a collection of triangular (or quadrilateral) contiguous, non-overlapping faces joined together along their edges. A mesh therefore contains vertices, edges and faces and its easiest representation is a single face. Sometimes it is also called TIN, Triangulated Irregular Network." [12]

Mathematical core of mesh is described in "Geometric Modeling Based on Polygonal Meshes", (Mario Botsch, 2008) In many geometry aproaches algorithms triangle meshes took in consideration as a cluster of triangles without any special mathematical structure. Each triangle defines as "barycentric parametrization, a linear segment of a piecewise linear surface representation.

Every point (p) in the interior of a triangle (a, b, c) can be written as a barycentric combination of the corner points:

$$p = \alpha * a + \beta * b + \gamma * c$$

with

$$\alpha + \beta + \gamma = 1$$

By choosing an arbitrary triangle (u, v, w) in the parameter domain, we can define a linear mapping (f): IR2 → IR3 with

$$\alpha * u + \beta * v + \gamma * w \mapsto \alpha * a + \beta * b + \gamma * c \qquad (5.1)$$

A triangle mesh M consists of a geometric and a topological component, where the latter can be represented by a graph structure (simplicial complex) with a set of vertices

$$\mathrm{V} = \{v_1, \dots, v_V\}$$

and a set of triangular faces connecting them

$$\mathrm{F} = \{f_1, \dots, f_F\}, f_i \epsilon \mathrm{V} \times \mathrm{V} \times \mathrm{V}$$

It is sometimes more efficient to represent the connectivity of a triangle mesh in terms of the edges of the respective graph

$$\mathcal{E} = \{e_1, \dots, e_E\}, e_i \epsilon \mathrm{V} \times \mathrm{V}.$$

The geometric embedding of a triangle mesh into IR3 is specified by associating a 3D position pi to each vertex $v_i \epsilon \mathrm{V}$:

$$P = \{p_1, \dots, p_V\}, p_i := (p_{\mathrm{U}_i}) =$$

such that each face $f \in \mathrm{F}$ actually represents a triangle in 3-space specified by its three vertex positions. The geometric embedding is defined by assigning 3D positions to the (discrete) vertices, the resulting polygonal surface is still a continuous surface consisting of triangular pieces with linear parametrization functions.” [13]

If a enough smooth surface is approached by such a linear function, the approximation error is of the order O(h^2), with h designating the maximum edge length. In considering of this quadratic approximation power, the error is descreased by a factor of 1/4 when halving the edge lengths. As this elaboration divides each triangle into four sub-triangles, it increases the number of triangles from F to 4F (Fig. 5.1). Hence, the approximation error of a triangle mesh is backward proportional to the number of its faces. “The actual magnitude of the approximation error depends on the second order terms of the Taylor expansion, on the curvature of the underlying smooth surface.” [13] From this we can decide that a passable approximation is possible with just a temperate mesh complication: The vertex density has to be locally converted to the surface band, such that flat areas are rarely sampled, while in banded places the sampling density is higher.

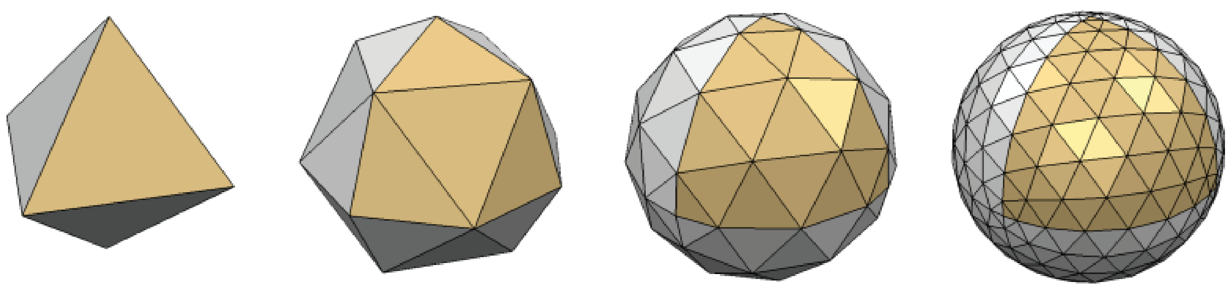

Figure 5.1: Each Subdivision Step Halves the Edge Lengths, Increases The Number of Faces by a Factor of 4, and Reduces the Error by a Factor of 1/4.[13]

As menshioned before, “an important topological quality of a surface is whether or not it is two-manifold, which is the case if for each point the surface is locally homeomorphic to a disk (or a half-disk at boundaries). A triangle mesh is two-manifold, if it does neither contain non-manifold edges or non-manifold vertices, nor self-intersections. A non-manifold edge has more than two incident triangles and a non-manifold vertex is generated by pinching two surface sheets together at that vertex, such that the vertex is incident to two fans of triangles (Fig. 5.2). Non-manifold meshes are problematic for most algorithms, since around non-manifold configurations there exists no well-defined local geodesic neighborhood.” [13]

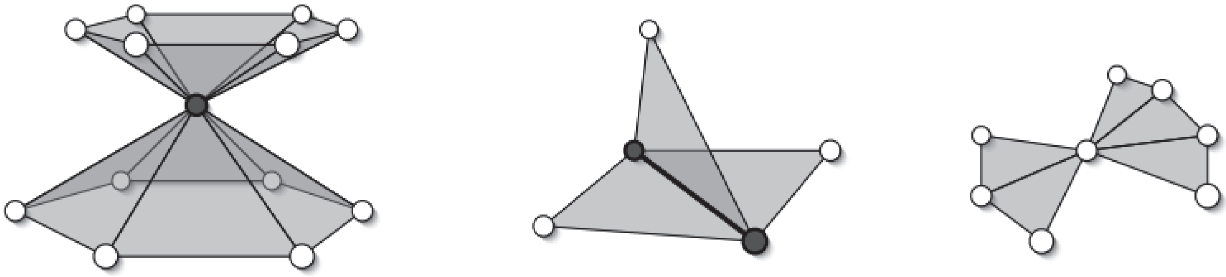

Figure 5.2: Two Surface Sheets Meet at a Non-manifold Vertex (left). A Non-manifold Edge Has More Than Two Incident Faces (center). The Right Configuration, Although Being Non-manifold in the Strict Sense, Can Be Handled by Most Data Structures.[13]

For piecewise surface determination, the most difficult part is the realization of smooth modulations between nearby patches. The triangle meshes have to make sure that nearby faces share a common edge. This makes polygon meshes the most simple and flexible surface figuring.

5.2. Creating a mesh

As laser scanners data usually is large array of data, it can cause a problem with processing the whole object file. Limits on file storage and transmission services make possible sharing only raw scans or point cloud or mesh which was created with low quality settings. But Faro Scene has a Clipping Box function and using this function the object was divided in 14 parts (Fig. 5.1).

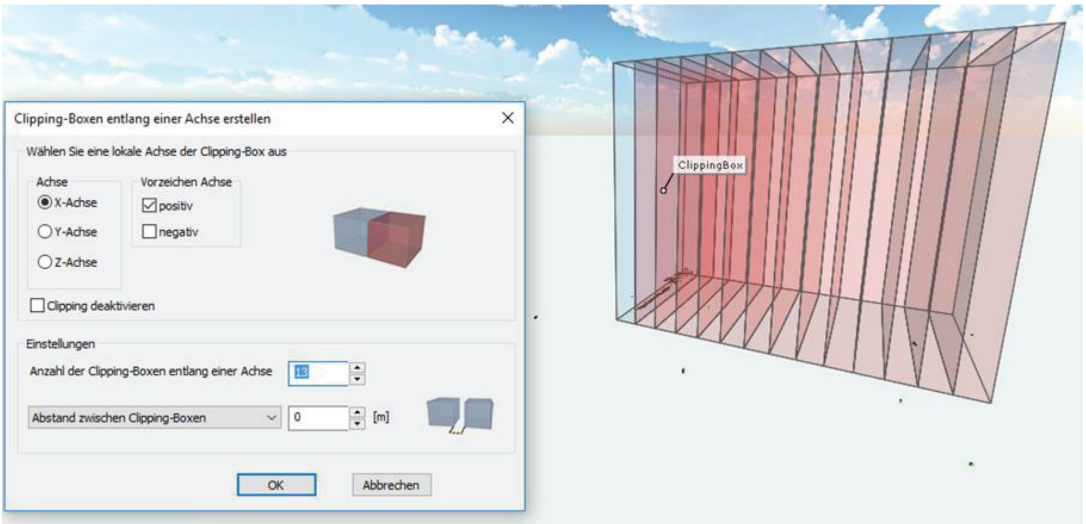

Figure 5.3: Setting Parameters To Multiplied Clipping Boxes.

To avoid the gap between boxes, zero was preset as distance between boxes. It is also possible to set distance between boxes centers, but it is not practicable. So this slicing allows to keep required PC resources for creating a mesh acceptably low. And in case of transmission data to some mesh modeling software it better to try initially with small part of data. For example if representation of the roofs surface should be compared, just some pieces of roof can be clipped and transferred in the different formats. After this for each "slice" a mesh was created. It was realized with standard settings, not watertight surface, none smoothing, geometry and colors were optimized and above middle number of triangles. "The watertight was not created because its takes a lot of PC source to create it. Watertight means that the mesh on all of the surfaces is complete, the lines of the mesh create valid elements, and the mesh properly connects to adjacent surfaces around the perimeter so that the volume is fully enclosed".[14] The requirement that volume always is enclosed causes wrong representation of the objects. In (Fig. 5.2) a watertight mesh is shown. This is seen from the back side, and the program created this "bubble" to enclose the mesh though it was not the right representation.

Figure 5.4: Watertight Mesh Surface

Watertight mesh is e.g. possible to use when just the interior representation is needed. When the object has to be represented outside it becomes impossible, because of that “bubble”. The difference of representation quality of these two kinds of surface is not visible to the human eye (Fig. 5.3). So the conclusion that a not watertight mesh is enough for transferring, can be made.

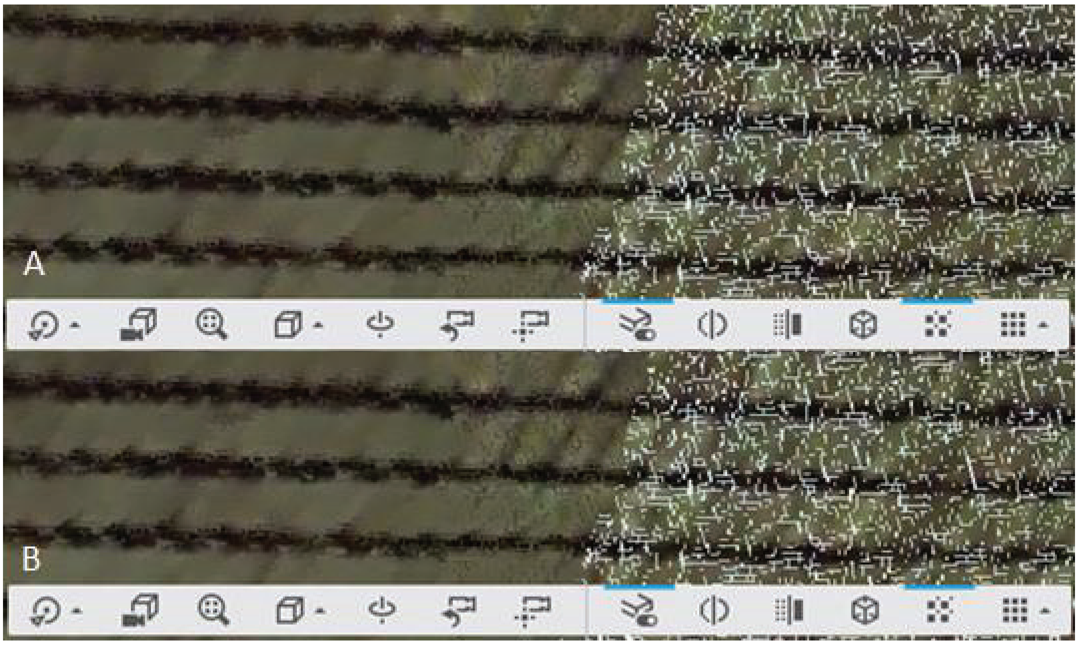

Figure 5.5: A – Not Watertight Mesh and Point Cloud, B – Watertight Mesh and Point Cloud

6. DATA TRANSFER FROM FARO SCENE TO BLENDER

6.1. Blender Software

Blender is a free and open source 3D creation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking up to video editing. There is the possibility to employ Blender's API for Python scripting to customize the application and write specialized tools; these APIs are included in Blender's releases.

Blender is cross-platform and runs equally well on Linux, Windows, and Macintosh computers. Its interface uses OpenGL to provide a consistent experience. To confirm specific compatibility, the list of platforms indicates those regularly tested by the development team.

As a community-driven project, the public is empowered to make small and large changes to the code base, which leads to new features, responsive bug fixes, and better usability.[15]

6.2. Types of used files

Faro scene can import point cloud data in the next file formats:

- E57 Files (*.e57)
- VRML File (*.wrl)
- DXF File (*.dxf)
- XYZ Ascii File (*.xyz)
- XYZ Binary File (*.xyb)
- IGES File (*.igs)
- PTS File (*.pts)
- Pointools POD (*.pod)
- CPE File (*.cpe)

Mesh data can be transferred and stored in:

- STL File (*.stl)
- OBJ File (*.obj)
- PLY File (*.ply)
- VRLM File (*.wrl/*.x3d)

Data can be imported in Blender Software in these formats:

- Collada (Default) (*.dae)
- Alembic (*.abc)
- 3D Studio (*.3ds)
- FBX (*.fbx)
- Motion Capture (*.bvh)
- Stanford (*.ply)
- Wavefront (*.obj)
- X3D Extensible 3D (*.x3d)
- Stl (*.stl)
- AutoCAD (*.dxf)

There are a number of additional file formats that blender supports, but these are often programmed by individuals and need to activated manually in the user preferences settings. Maintenance of these import/export plugins is not always ensured for new Bender releases. For that reason these file formats are not investigated in this theses.

The essential workflow to get the laser scanned data into the mesh modeling software is depicted in figure 6.1:

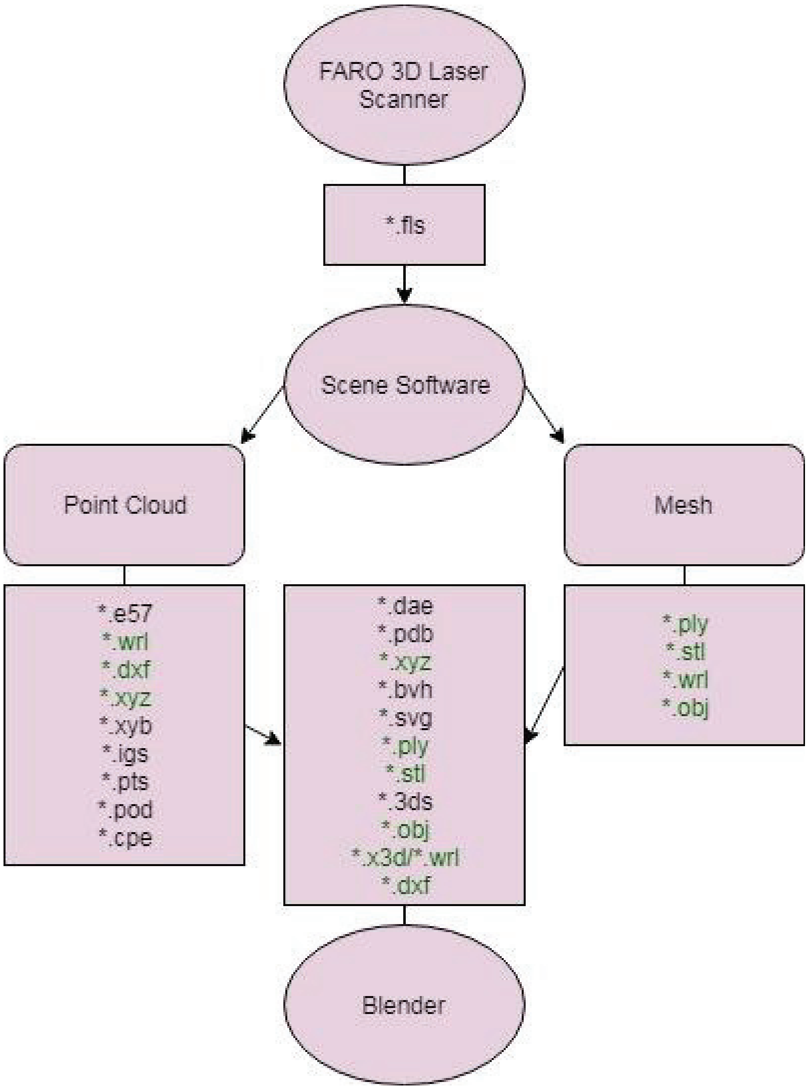

Figure 6.1: Data Transfer Scheme

6.2.1. Stanford file format

In “An implementation of 3D point based rendering system”,(Peter Cracknell, 2005) author gives next explination of Stanford or PLY format “PLY format describes an object as a collection of vertices, triangles and other elements, along with properties such as color and a normal direction that can be attached to these elements.

A typical PLY file has the following structure:

- Header: a series of carriage-return terminated lines of text that describe the remainder of the file. It includes a description of each element type, including the element's name, how many such elements are to be found in the object, and a list of the various properties associated with the element.

- Vertex List: a list of vertices, corresponding to the header's definition
- Face List: a list of triangles, usually containing an unsigned char specifying how many indices the property contains, followed by the actual list of vertex indices stored as long integers".[16]

According to the (Greg Turk, 1998) the PLY format is not designed to be a general scene description language, a shading language or a all-out modeling format. This means that it includes no transformation matrices, object examolples, modeling theocracy, or object subdivisions. It does not include parametric patches, quadric surfaces, CSG technology, triangle strips, polygons with holes.

A usual PLY object definition is a list of (x,y,z) coordinates for vertices and a list of faces that are indicated by indexes into the list of vertices. Most PLY files include this base information. PLY file is its list of elements. Each element in a existing file has a established number of attributes that are specified for each element. The standart information in a PLY file contains just two elements, the (x,y,z) coordinates for vertices and the vertex index for each face. Programs can add new properties that are attached to elements of an object. Color properties are commonly associated with vertex elements.

New attributes are attached in a way that old programs do not ruin when these new properties are collided. Attributes that are not clear by a program can either be accepted unspent or can be rejected. It is possible to create a new element type. The attributes associated with this element are then to be defined. Examples of such new elements are edges, cells and materials (ability to reflect the light).

Limitations of the PLY format:

- PLY is designed as a simple, easy to analyze file format and from here only express basic geometry information.

- Only one object definition can be specified per PLY file. If there is more than 1 object in a 3D scene, more than one PLY file have to be exported.

- No material information are stored in the PLY format.

- No lights, cameras are provided by the PLY file format.[17]

Figure 6.2: Imported mesh in PLY format in Blender

6.2.2. STL file format

One of the most common file formats is the .STL file. The file format to have been created by 3D Systems from its STereoLithography CAD software and machines.

“The STL file format cannot store additional information such as color, material etc. of the facets or triangles. It only stores information about the vertices and the normal vector.” [18]

Like many file formats, there are other explanations for how this file-type received its name: Standard Tessellation, which shall imply tiling or layering of geometric shapes and patterns. But there is no such thing like a ‘standard’ tesselation. In algorithmic geometry and contemporary computer science there are various tesselation approaches.

An easy-to-understand definition of the STL gave (TJ McCue, 2018) author explained STL as a triangular representation of a 3D object. If you look at a CAD file, it shows smooth lines for circles, where an STL file shows the surface of that circle as a series of connected triangles.

This is used to represent 3D printer drawings as mesh files – because they are not solid but made up of triangles creating a mesh or net-like appearance.

Several factors determine the quality of an STL file:

- Chordal Tolerance / Deviation

  This is the distance between the surface of the original object and the mosaic of triangles.

- Angle Control

  It is possible to have gaps between triangles, and changing the angles (deviation) between adjacent triangles will improve the print resolution. This setting enables to increase how close objects are layered or tiled together.

- Binary or ASCII

  Binary files are smaller and easier to handle. ASCII files have the advantage of being easier for reading, visualization and checked.[19]

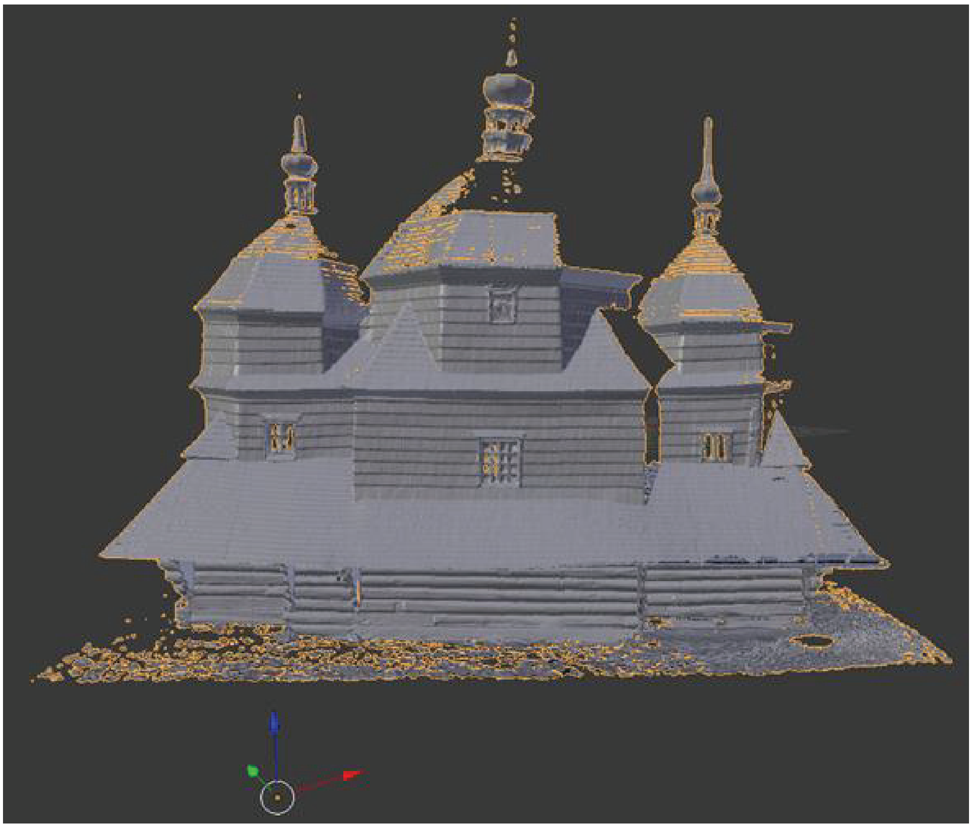

Figure 6.3: Imported mesh in STL format in Blender

6.2.3. Virtual Reality Modeling Language file format

The Virtual Reality Modeling Language (VRML) is a file format for representation interactive 3D objects. VRML was designed to be used on the Internet, and local network systems. VRML is also designed to be a universal exchange format for integrated 3D graphics and multimedia. VRML may be applied in a many of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages.

VRML is able of expressing stable and active 3D and multimedia objects with links to other media such as text, sounds, movies, and images. VRML browsers, as well as authoring tools for the creation of VRML files, are widely available for many different applications.

“An additional feature is that VRML supports an extensible model that allows new dynamic 3D objects to be defined allowing application communities to develop interoperable extensions to the base standard. There are mappings defined between VRML objects and commonly used 3D application programmer interface (API) features.” [20]

According to (ISO, 1997) VRML files can store the following entities and features:

- Camera views
- Edges
- Faces
- Groups
- Lights
- Materials and textures
- Transparency

A VRML file consists of the following major functional components:

- header,
- scene graph,
- prototypes, and
- event routing.

The scene graph describes objects and their properties. It contains hierarchically grouped geometries to provide an audio-visual representation of objects or to nodes participate in the event generation and routing mechanism.

“Prototypes allow the set of VRML node types to be extended by the user. Prototype definitions can be included in the file in which they are used or defined externally. Prototypes may be defined in terms of other VRML nodes or may be defined using a browser-specific extension mechanism. While ISO/IEC 14772 has a standard format for identifying such extensions, their implementation is browser-dependent.

Some VRML scene graphs can generate events in response to environmental changes or user interaction. Event routing gives authors a mechanism, separate from the scene graph hierarchy, through which these events can be propagated to effect changes in other nodes. Once generated, events are sent to their routed destinations in time order and processed by the receiving node. This processing can change the state of the node, generate additional events, or change the structure of the scene graph." [21]

Figure 6.4: Imported mesh in VRML format in Blender

6.2.4. Wavefront file format

Wavefront OBJ is a ASCII file format for showing 3D mesh geometries such as position of vertices, texture coordinates, edges and the faces of each polygon defined as a list of vertices. OBJ files can not content any animation, object hierarchy, transformation or even the use of 3D figures as spheres and cubes. Information about the material can also be stored in OBJ files, however can be tied in MTL files using OBJ's Material Template Library (MTL) companion file format. Faro Scene does not provide MTL extensions for OBJ files, so the color information is not exported.

Despite these limitations, the OBJ format by is in the Computer Graphics community a widely accepted format and remains in common use due to its relative simplicity. The MTL file format, however, is much less supported, so transformations between different 3D graphics programs brings to result in a loss of material and color information.[22]

The OBJ file format supports both polygonal objects and free-form objects. Polygonal geometry uses points, lines, and faces to define objects while free-form geometry uses curves and surfaces.

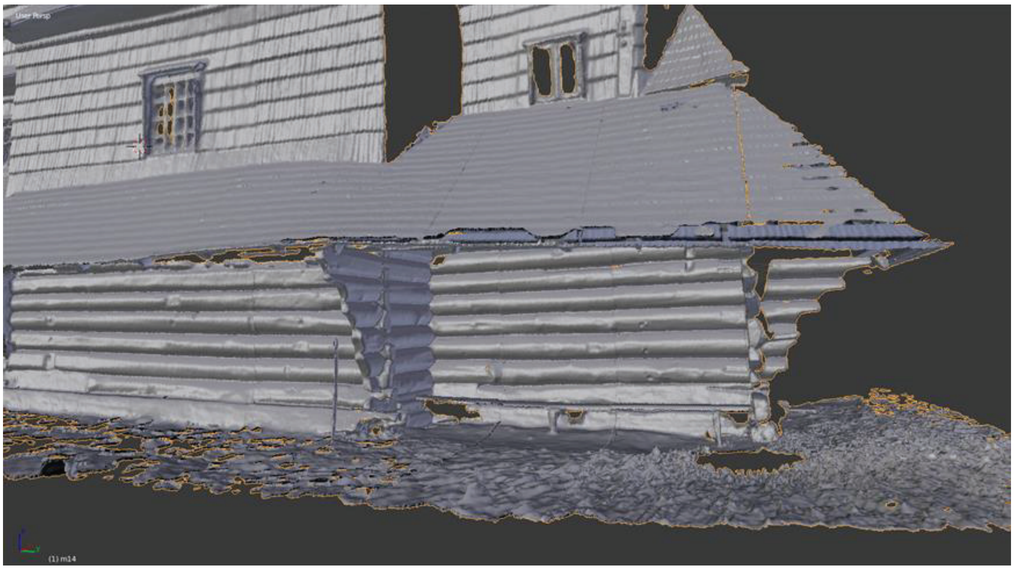

Figure 6.5: Imported mesh in OBJ format in Blender

6.2.5. AutoCAD file format

“A file with the .DXF file extension is a ‘Drawing Exchange Format’ file developed by Autodesk as a type of universal format for storing CAD models. The idea is that if the file format is supported in various 3D modeling programs, they can all import/export the same documents with ease.

The DXF format is similar to the AutoCAD Drawing Database file format that uses the DWG file extension."[23] However, DXF files are more widely used for data exchange since it can encode geometry information in a text-based, ASCII format that naturally makes it easier to implement in these types of applications.

"Every variable in the DXF file is joined with a number of codes (from 1 to 1071); each group code is used in a definite context and contain specific types of information. For example, group code 2 is used for names, such as names of sections or names of blocks. Group code 0 refers to the start of an entity or the end of the file.

The values associated with each variable are stored as an integer, floating-point number or a string. For example, the length of a line is stored as an integer, while point coordinates are stored as an integer, while point coordinates are stored as a floating point number.

A DXF file is organized into several sections; each section is composed of records, which in turn are composed of a group code and associated data values. Here are the sections that DXF file contents."[24]:

- Header
- Classes
- Tables
- Blocks
- Entities
- Objects

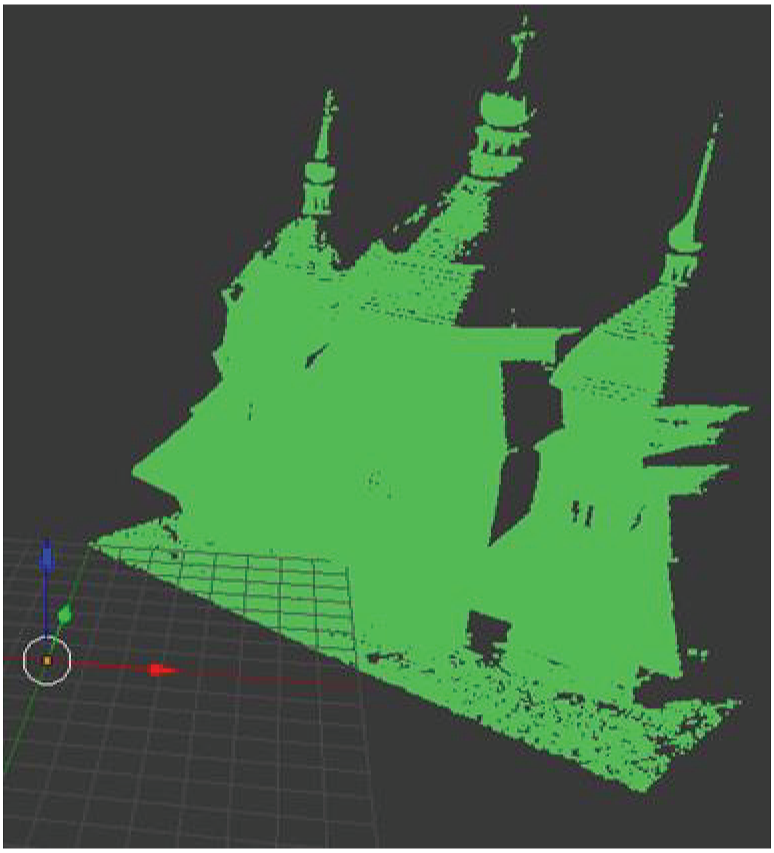

Figure 6.6 Imported mesh in DXF format in Blender

6.2.6. XYZ file format

"An XYZ file is either an ASCII file or a binary file that contains a set of vertices. When reading a XYZ file, if the optional header is not used, a dialog box will appear so that required information can be entered by the user.

A set of three floating point numbers represents the X, Y and Z coordinates of a vertex. In an ASCII file, the decimal point only has to be written if it is necessary. A binary file can be in either Intel and the floating point numbers can be either floats (4 bytes) or doubles (8 bytes)".[25]

Importing of XYZ files in Blender demands installing additional utilities. There are two ways to import this kind of files first is XYZ –Atomic Blender. This utilities allows to import direct XYZ file in Blender. The another way is using *CSV Mesh Importing*. This can import a CSV file and create meshes, such as vertices, edges and faces. This can import two types of CSV files: point list containing X, Y, Z columns and X-Y point 2D table (matrix).

7. CONCLUSIONS

To estimate the most suitable format to use in data transferring some important factors were highlighted:

- Ability to transfer information about the vertex color
- Orientation of axis
- Units
- Size of file

The orientation of the axis shows if file format keeps the same orientation which object got during laser scanning. In case of using Blender software it is not so important, because it can be easily changed. But if consider possibility to work with transferred data in other software packages, it becomes important. When a lot of parts of one object or a lot of objects should be connected in one file it is more fast and convenient when they keep the same axis orientation.

“Blender uses five types of units systems Metric, Imperial, Radians, Degrees and system with Blender Units. Blender Unit is used as default unit system ans it has no real equivalent.” [26] Faro Scene supports only metric system, so the question was to check does metric system stay after importing in Blender. In case of just representing the object it is not so important, but in case of doing some researches or measuring on it object it demands real size of object. Also it is necessary during importing different files in one project

Size of file: Processing of laser scanning data usually means handling with a big size files, so file size is important parameter.

Table 7.1

Comparing Of Imported Mesh Files

	Color	Orientation	Units	Size, MB
PLY	+	X=0° Y=0° Z=0°	Meters	168, 173
STL	-	X=0° Y=0° Z=0°	Meters	314, 773
VRML	+	X=90° Y=0° Z=180°	Meters	438, 834
OBJ	-	X=0° Y=0° Z=0°	Meters	319, 431

STL file does not contain color information as it is a file format for geometrically oriented devices like 3D printers. Using this STL would be possible only together with

UV mapping. UV mapping it is projecting images on 3D objects, what allows to show the textures and colors. This mapping is usually done in the 3D-modeling software package. Blender contains a number of usefull algorithms to perform this task effectively.

OBJ file itself has not colors. Colors can not be exported form Faro because Faro Scene does not create an MTL file. Without MTL file which contents all texture information OBJ shows only the shape of the object.

PLY file format appears to be the most practicable among the presented mesh formats. It has texture information and keeps the scanner orientation. The size of the exported church file is two times smaller than STL and OBJ files.

VRML file has color information, but the size is bigger as PLY and the exported model has a different orientation than the scanner. Information from Faro Scene can be stored as point cloud in VRML format. But again, VRML has a deviating orientation, which would have to be corrected in the mesh modeling software, and the size of point clouds is many times bigger in the mesh data. And what is more important, Faro Scene does not store the information about color in VRML.

The DXF file has the correct orientation, but the size is almost 10 times bigger than that of the PLY file.

Table 7.2

Comparing Of Imported Point Cloud Files

	Color	Orientation	Units	Size, MB
DXF	-	X=0° Y=0° Z=0°	Meters	1 465, 355
VRML	-	X=90° Y=0° Z=180°	Meters	1 398, 302
XYZ	-	X=0° Y=0° Z=0°	Meters	899, 506

In case, the XYZ format shall be used, it is recommended to be converted first in Excel CSV file and than using *CSV Mesh Importer* in Blender.

Figure 7.1: Scheme Of Importing XYZ Point Cloud In Blender Using CSV Mesh Importing

Point clouds usually contain millions of points but an Excel file can have only 1 048 576 rows. So it can be used only for small objects (Fig. 7.2). But XYZ has the advantage in size comparing to other point cloud formats. The Size of XYZ file is almost two times smaller than DXF or VRML. It is also possible to create mesh from point cloud inside Blender using *Point Cloud Skinner.* But this instrument works very slowly. The piece of point cloud was processed in 12 minutes with minimal settings, the size of that piece was 83 KB. Given this, this utility can not be considered as usable tool.

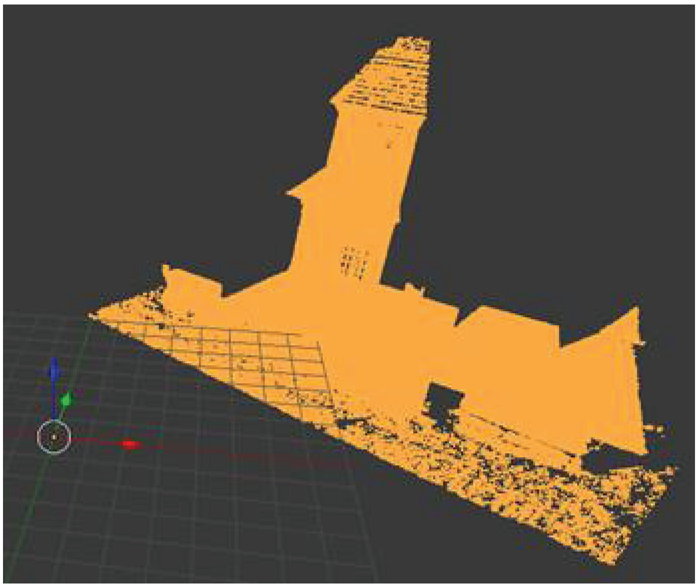

Figure 7.2: Imported XYZ Point Cloud In Blender Using CSV Mesh Importing

Based on this result, mesh data is certainly a more convinient format than point cloud data for transferring into Blender software for further visual representation. Stanford file format is the most practicable choice for presenting the view of stable objects. It is suitable for historical buildings and etc., as it has color information and its compression algorithm worked better comparing to other possible file formats.

Regarding to this pipeline of data transfering from terestric laser scans can be defined (Fig 7.3).

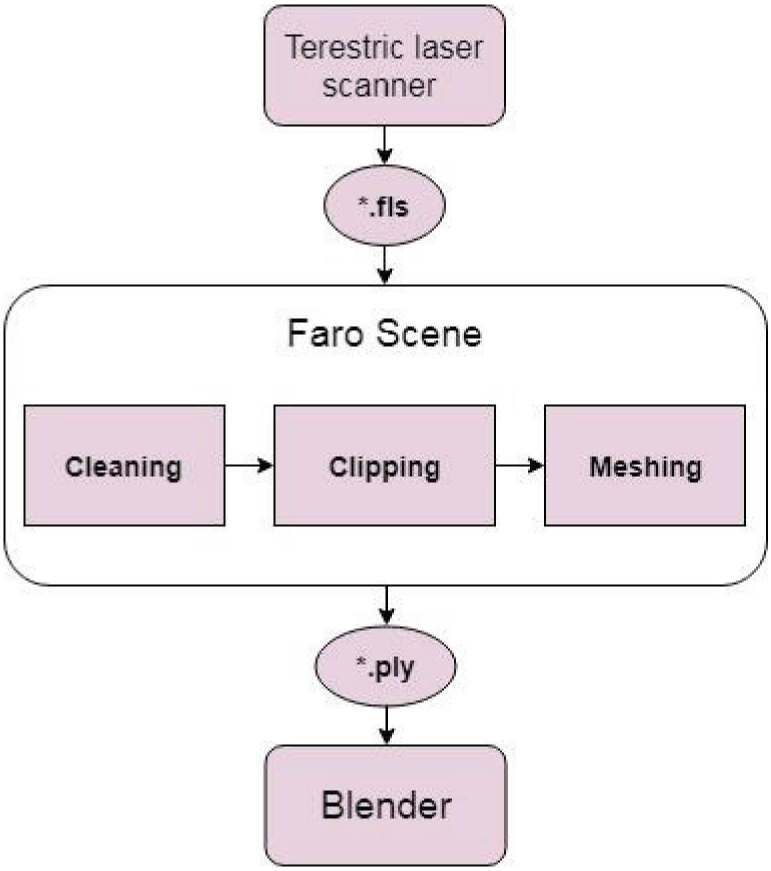

Figure 7.3 Pipeline of data transferring

The part of transferring data into Faro Scene is defined, because it is the only possible way provided by Faro. After routine processing data in Faro Scene data should be cleaned in scans or in point cloud, if necessary “sliced” and then a mesh should be created. The most practicable way to transfer data from Faro Scene to Blender is to use the PLY format.

8. REFERENCES

[1] Romanishin, A. Malitskyy, A. Lozinski (2012): Classification and main features of 3D-surface scanners; in: Geodesy and geodynamic, volume II (24), pp. 69-74.

[2] Norbert Pfeifer, Christian Briese (2007): Laser scanning – principles applications; in Conference: GeoSiberia 2007 - International Exhibition and Scientific Congress, p.7.

[3] Boehler, W., Heinz, G., Marbs, A. (2001): The potential of noncontact close range laser scanners for cultural heritage recording, XVIII. CIPA Symposium, Potsdam, Germany, Proceedings. p. 2.

[4] Theory and practice on Terrestrial Laser Scanning (2008), p. 22.

[5] Practical application of 3D laser scanning techniques to underground projects (2012): A part of ISRM-Swedish national task: "A survey of 3D laser scanning techniques for application to rock mechanics", Stockholm.

[6] FARO® Laser Scanner Focus3D Manual (October 2011)

[7] Wikipedia (2018): Temple St. Archangel Michael (Komarno); in: Храм_Архістратига_Михаїла_(Комарно) (18.04.2018)

[8] FARO Technologies (2018): FARO Scene; in: https://www.faro.com/products/product-design/faro-scene/.

[9] Anne Bienert, Hans-Gerd Maas (2009): Methods For The Automatic Geometric Registration Of Terrestrial Laser Scanner Point Clouds In Forest Stands; in: : Bretar F, Pierrot-Deseilligny M, Vosselman G (Eds) Laser scanning 2009, IAPRS, Vol. XXXVIII, Part 3/W8

[10] J. L. Lerma and D. García-San-Miguel (2014): Self-Calibration of Terrestrial Laser Scanners: Selection of the Best Geometric Additional Parameters, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-5

[11] P.-F. Käshammer and A. Nüchter (2015): Mirror identification and correction of 3D point clouds, in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-5/W4

[12] Remondino Fabio (2004): From point cloud to surface: The modelling and visualisation problem, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV-5/W10

[13] Mario Botsch (2008): Geometric Modeling Based on Polygonal Meshes, in: Eurographics 2008 Full-Day Tutorial.

[14] Autodesk (2017): Watertight problems in meshes; in:
https://knowledge.autodesk.com/support/simulation-mechanical/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/SimMech-UsersGuide/files/GUID-760A8862-E798-4BC2-AA56-BCA611B0EB7F-htm.html
[15] Blender (2018): The software; in: https://www.blender.org/about/
[16] Peter Cracknell (2005): An implementation of 3D point based rendering system, in: Mathematics and Computing (May, 2005)
[17] People.sc.fsu (1998): The PLY Polygon File Format; in:
https://people.sc.fsu.edu/~jburkardt/data/ply/ply.txt
[18] ALL3DP (2017): STL File Format for 3D Printing – Simply Explained; in:
https://all3dp.com/what-is-stl-file-format-extension-3d-printing/#pointten
[19] Lifeware (2018): STL Files: What They Are and How to Use Them
https://www.lifewire.com/stl-files-2255
[20] SketchUp Help Center (2017): Exporting VRML Files; in:
https://help.sketchup.com/es/article/3000174b
[21] International Standard ISO/IEC 14772-1:1997
[22] Adrewonske (2015): OBJ file format; in:
http://www.andrewnoske.com/wiki/OBJ_file_format
[23] Lifeware (2018): What is a DXF File?; in: https://www.lifewire.com/dxf-file-4138558
[24] Scan2CAD (2016): The DXF File Specification; in:
https://www.scan2cad.com/dxf/file-specification/
[25] MicroMouse (2018): XYZ Files; in:
http://www.micromouse.ca/xyz_files.html
[26] Wiki.Blender(2018): Units; in:
https://wiki.blender.org/index.php/User:Rayek/Doc:2.6/Manual/Interface/Units