

Guideline

Document name

Content production guidelines: Laser scanning



Project Acronym:	IMMERSIFY
Grant Agreement number:	762079
Project Title:	Audiovisual Technologies for Next Generation Immersive Media

Revision:	1.1
Authors:	Maciej Jaśkiewicz Jacek Walczak Maciej Głowiak
Support and review	Maciej Stróżyk Szymon Malewski
Delivery date:	M30
Dissemination level (Public / Confidential)	Public

Abstract

This document describes how to acquire 3D data with laser scanners into point clouds and briefly presents the available formats and software. In addition, it describes the possibilities of editing and post-production of point clouds for high-resolution video production in various formats: 8K, 16K, 3D, 360 along with a discussion of projection types such as ERP (Equirectangular Projection) or CMP (CubeMap Projection). It also discusses the preparation and encoding of the final image using software produced by the Immersify project. The document contains practical experience gained during the scanning of the cathedral in Poznan.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 762079.

TABLE OF CONTENTS

1	4
2	4
2.1	4
2.1.1	4
2.1.2	4
2.1.3	5
2.2	7
2.2.1	7
2.2.2	7
2.3	8
3	9
3.1	9
3.2	9
3.3	9
3.4	10
3.5	11
3.6	12
4	12
4.1	12
4.2	12
4.2.1	12
4.2.2	12
4.2.3	13
4.3	14
4.3.1	14
4.3.2	15
4.3.3	15
4.3.4	16
4.3.5	17
4.4	18
4.5	18
5	20
5.1	20



5.2 21

5.2.1 21

5.2.2 21

5.2.3 22

1 Introduction

Due to technological constraints, the acquisition of high-definition video is currently limited. While 4K technology is becoming more and more popular in both professional (cinema) and home (television, mobile phones) applications, still there are only a few, prototype solutions for obtaining 8K content. 360° cameras are also limited to 8K, but this resolution applies to full spherical representation, which means that the effectively watched resolution on HMD is much lower.

In order to generate photorealistic video content with a higher resolution, in practice we have 2 options:

- use methods of designing and rendering 3D objects (CGI - Computer Generated Imagery)
- acquire content with the use of 3D scanners (point cloud)

The purpose of this document is to describe best practices of the second approach, with acquisition, editing, processing, animating and rendering

2 Acquisition

2.1 Introduction

The first step towards the production of photorealistic images based on the point clouds rendering is to record raw data acquired by the laser scanner or LIDAR (Light Detection and Ranging).

2.1.1 *Point clouds*

Point cloud is a dataset containing information about points in three-dimensional space. For each point in the collection, XYZ coordinates are stored as well as colour assigned to every single point. Data can be stored in miscellaneous formats .e57, .ptx, .las, .laz, .xyz, .pts, .rcp, .rcs and others but basically they include X-axis coordinate, Y-axis coordinate, Z-axis coordinate and RGB values of each point in the cloud.¹

Point clouds are used in many fields of technology and application areas. Objects described by point clouds can be as large as big cities, including buildings, roads with cars and trees - or as small as a few millimetres (e.g. coin). Point clouds are usually the basis for highly accurate 3D models, which are then used for calculations and measurements directly in or on the object (e.g. dimensions, distances or cubatures). They are therefore a great source of information for 16K renderings, 360° videos or other CGI techniques.

2.1.2 *Scanning devices*

Laser scanners are mainly used to create point clouds, and there are many various models available on the market, but the most frequently used LIDARs are produced by FARO, Leica Geosystems and

¹ [Wikipedia - Point cloud](#)

Artec3D. Differences are mostly in range, depth, metadata formats, imaging and imaging range but the general idea and working principle is always similar.

In the first step the device calibrates itself and receives metadata about its location (e.g. GPS data). When the position of the scanner is precisely defined, it starts sending out a laser beam and then examines the light reflected from the objects on the path of light. The diffuse luminous intensity value is converted into the corresponding coordinates of the points in space and saved in the defined format of the point clouds datasets. Finally, the scanner measures the exposure and takes pictures to obtain information about the point colours.

We use **FARO Focus 3D X330** laser scanner (Fig. 1) for all of our scans. It can scan objects from 0.6 meter up to 330 meters away even in direct sunlight. With its integrated GPS receiver, the laser scanner is able to correlate individual scans in post-processing. It measures up to 976,000 points per second and integrates colour camera with up to 165 Million pixels. It may be controlled via touchscreen display or WLAN.



Figure 1. FARO Scanner unboxing

2.1.3 Scanning preparation procedure

The most important thing before starting a scan is to plan the scanner's positions in the scanned object. In order to do it properly, it is required to understand how a 3D scanner works and what are its limitations.

The scanner emits a laser beam around its position in all directions, so one single scan will collect information only about the front sides of the surrounding environment (Fig.2). To get more detailed information about the whole object shape, you have to take another scans from different positions, for example from the backside of the object (Fig. 3). Planning the position, take the height also into account - scanning reliefs, statues or shelves requires a tripod with various height settings. (Fig. 4).

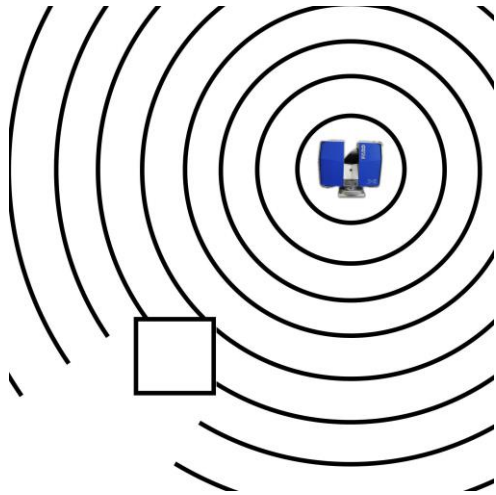


Figure 2. Single scan example.

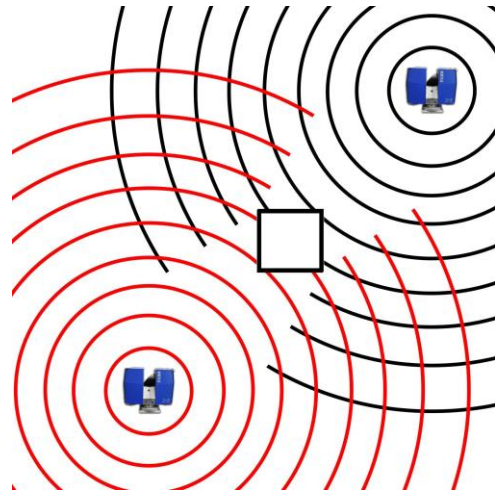


Figure 3. Multiple scan example.



Figure 4. FARO scanner on high tripod.

Two most important settings during the scanning are: **profiles** that match the scanning conditions (*Indoor, Outdoor, Sunlight, etc.*) and **scope of operation** (*Whole sphere or Slice of the sphere*). Additionally you can choose **Black&White (BW)** or **Color** scan and set **quality & resolution** parameters of the scan. The higher quality and resolution is set the longer period of time is required for a single scan. Following practical examples show the differences of single scanning time depending on scanner settings:

- **Resolution 1/8, quality 6x:** scan size 5120x2134 pts (11M pts), point distance 12.272mm/10m - scan time **11 min 22 sec**
- **Resolution 1/4, quality 3x:** scan size 10240x4267 pts (44M pts), point distance 6.136mm/10m - scan time **7 min 47 sec**
- **Resolution 1/2, quality 3x:** scan size 20480x8534 pts (175M pts), point distance 3.068mm/10m - scan time **18 min 32 sec**

Most scans in our tests (see below) used: **Resolution 1/8** and **quality 4x** (scan size 8120x3414 pts, 28M pts, point distance 7.67mm/10m) with the scan time: **8 min 48 sec.**

2.2 Scanning the Cathedral church

2.2.1 *The history of the church*

The cathedral was originally built in the second half of the 10th century, and in the 14th and 15th centuries the church was rebuilt in the Gothic style. At that time, a crown of chapels was added. A fire in 1622 did such serious damage that the cathedral needed a complete renovation, which was carried out in the Baroque style. Another major fire broke out in 1772 and the church was rebuilt in the Neoclassical style. In 1821, Pope Pius VII raised the cathedral to the status of a Metropolitan Archcathedral. The last of the great fires occurred on 15 February 1945, during the liberation of the city from the Germans. The damage was serious enough that the conservators decided to return to the Gothic style, using as a base medieval relics revealed by the fire. The cathedral was reopened on 29 June 1956. In 1962, Pope John XXIII gave the church the title of Minor Basilica.²

Today, the cathedral is a three-nave basilica church, with a farmyard around the chancel. There are 12 chapels and 2 sacristies adjacent to the side aisles and the farmyard. The temple is 81 m long and 43.5 m wide, the height of the main nave is 24.5 m, the towers are approx. 62 m, and the turrets above the farmyard are 44 m.

2.2.2 *Scanning*

In order to produce high quality photorealistic images of such complex architectural objects as the Cathedral it was required to prepare a detailed plan and sequence of the scanning and select appropriate places for scanning based on the photo documentation and available plans of the object. We took into the account that the final point cloud was to be used to render the virtual camera's passage through the church interior.

Scanning took a total of 3 days. On the first day, we made a local vision, looked at the church and pre-determined the scanning locations by applying them to the plan of the cathedral. The positions of the scanner during the Cathedral scanning procedure are marked with red dots and are depicted on Fig. 5. We had to take into account the covered architectural elements such as pillars, altars, side chapels, chairs, figures of saints, etc. The main idea was to scan only the interior of the temple and the golden chapel, so we did not take into account the scans from the outside (a drone should be used here), towers, additional rooms such as the sacristy or storages and the undergrounds hiding the Romanesque remains of the cathedral and crypt of deceased bishops and kings.

The next day passed on scanning the Golden Chapel and the main nave. A total of 56 scans were made, of which 48 were qualified for further processing. After an additional day of scanning, the total number of scans was 74, of which 65 were kept for further processing at the end. Each scan lasted about 25 minutes, of which 15 minutes took to move the scanner, verify its position and launch, while the scan itself took about 10 minutes. Each of the scans brought in about 30 million new pixels, which in total

² [Wikipedia - Poznan Cathedral](#)

gave us above 2 billion points. In total scanning of the interior of the cathedral took 25 hours and the resulting point cloud was about 38 GB of data. In the next stage of merging the received point clouds, the number of points and the size of the resulting cloud was obviously significantly reduced, because after merging many points were unified and removed. The final point cloud contains 1.4 billion points which took 24 GB.

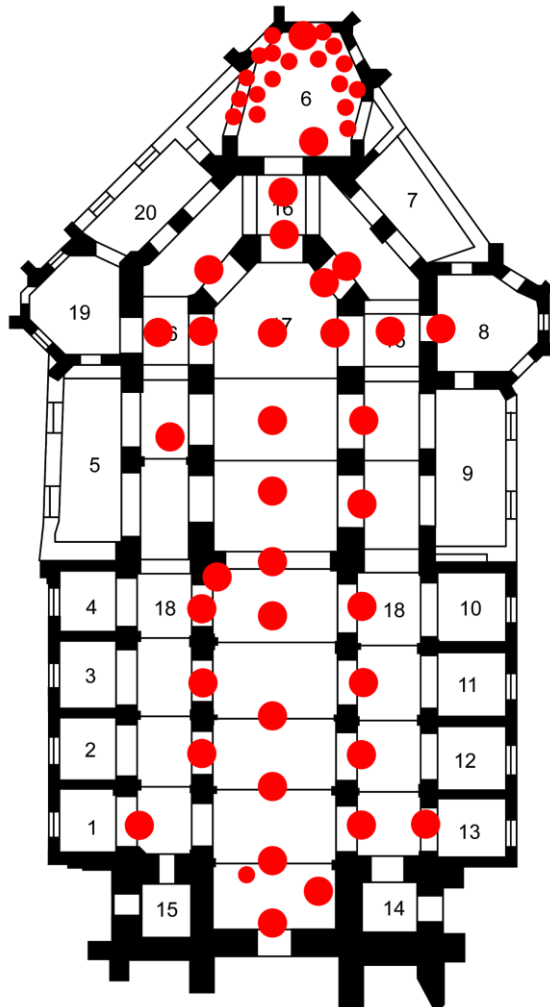


Figure 5. Plan of the cathedral church with positions of the scanner

2.3 Summary and good practices

- **Planning** - all organizational aspects of the scanning should be well planned in advance. There are many issues that should be taken into consideration before you start, like a scanning schedule that takes into account time when there are no people or moving objects inside the scanned building or constant light conditions during the whole process, etc. Important part of the planning process is also selection of the scanner placing points appropriate to cover the whole space and all objects regarding the height and distance from walls, etc. In this case very useful is detailed photo documentation.
- **Scanner calibration** - wait until the scanner receives GPS coordinates and do not move it afterwards. It will affect scans, lowering precision or making them unusable. If GPS is unavailable e.g. indoor places stitching will be harder, but is still possible.

- **Number of scans** - take more than one scan at given position, if there are moving objects (e.g. people and animals).
- **Open spaces** - in big, open spaces without any distinctive signs, use markers to make the stitching possible.
- **Control exposition** - try to avoid variable lighting conditions during the scanning process, always make sure that all scans have similar light conditions.
- **Avoiding reflecting surfaces** - Cover the windows and all the reflecting surfaces, like mirrors or metal elements. To make reflecting objects as non-reflecting use cyclodecane spray. It simply vanishes by evaporation (sublimation) after a few days, therefore, it does not harm the surface.
- **Avoiding moving objects during scan** - people walking around or other objects in move will cause shadows and scanning errors that are difficult to find and remove, especially in scans with large amounts of data. It is still possible during editing of the point cloud, however the process takes a lot of time and must be done carefully.

3 Editing

3.1 Introduction

Acquisition of the raw data by the scanner is only the first step towards photorealistic images generated from the point clouds. The second step is to combine individual scans into a single point cloud. This process, called **stitching**, is a main part of the post-production editing process. The other is removing *ghosting* – effect of objects scanned in move (such as people walking around).

3.2 Stitching Software

In general, the scanner manufacturers provide dedicated, commercial software for stitching and editing scans from their scanners. There are various applications dedicated for different scanners e.g. Leica CYCLONE - for Leica Geosystems scanners or FARO SCENE for Faro scanners, but there is also software produced by third-party companies like ReCap by Autodesk³ (ca. \$330 per year) or even open source software like CloudCompare⁴ (free of charge). In our work, we mostly used FARO SCENE⁵ software for stitching and editing scans, as it integrates seamlessly with FARO Laser Scanner. In our case the open source CloudCompare was selected only for animation purposes, but it may also be used for stitching and editing.

3.3 Stitching procedure

Stitching is a key process affecting the quality of created point clouds and it depends on the proper placement of scans inside the cloud and the correct matching of overlapping fragments of scans. In order to process the stitching in a proper way, it is required to record individual scans with appropriate

³ [Reality capture and 3D scanning software for intelligent model creation](#)

⁴ [CloudCompare](#) open source project

⁵ We used FARO SCENE in version 5.4, but the newest available is 7.1

(approximately 20%) set of overlapping points and surfaces. Scans without a common set of points cannot be stitched. Using any stitching software, all individual scans should be imported into one point cloud project and all the unnecessary points should be deleted to remove glitches.

Below, there is an example of how to do this using FARO SCENE software in a few simple steps:

Step 1 - Create project and import scans

Simply import the entire directory with the FLS files generated by the scanner and all FLS files into the new scene workspace.

Step 2 - Preprocessing

In this step you can verify your scans and use photo images from the scanner if you want to create a colourful point cloud. The most popular scanners on the market, including FARO, are able to automatically take colour photos during the scanning and you can then use these photos to apply colours and textures to the generated point clouds. To do this just export the photos in proper (e.g. equirectangular) format with the required resolution (e.g. in our case it was about 8,120 x 3,414 pixels) and processed if required (e.g. blur some logos or faces). The images prepared in this way can be directly included in the point cloud workspace.

Step 3 - Scans placement

Scans can be placed in the points cloud in various ways (manually or automatically), usually the automatic algorithms work quite well and the final result is satisfactory if the overlapping areas are large enough. In case the individual scans are less accurate or the overlapping regions are too small for automatic mode, the manual mode based on the average distance between points or selected reference points can be applied.

Step 4 - Stitching precision adjustment

When the scans placement procedure is finished the window with all detailed information including precision of stitching will appear. In general, a well-planned scanning process means that more than 70% of the points are closer than 4mm. In case of potential problems just update scans, change mode or parameters.

3.4 Removing objects

An important part of point cloud processing is the procedure of removing unwanted objects and fixing any other glitches. Most commercially available stitching software, including FARO SCENE offers tools for semi-automatic or manual removing and cleaning the final point cloud. Most commonly we will encounter the following types of distortion:

Far objects

Because of the long range of scanner laser light, a common case is that the laser beam gets out through the windows or we capture an object that is too far away from the point of interest. We should remember to limit the final range of the points. Fig. 6 depicts artefacts of far glitches.

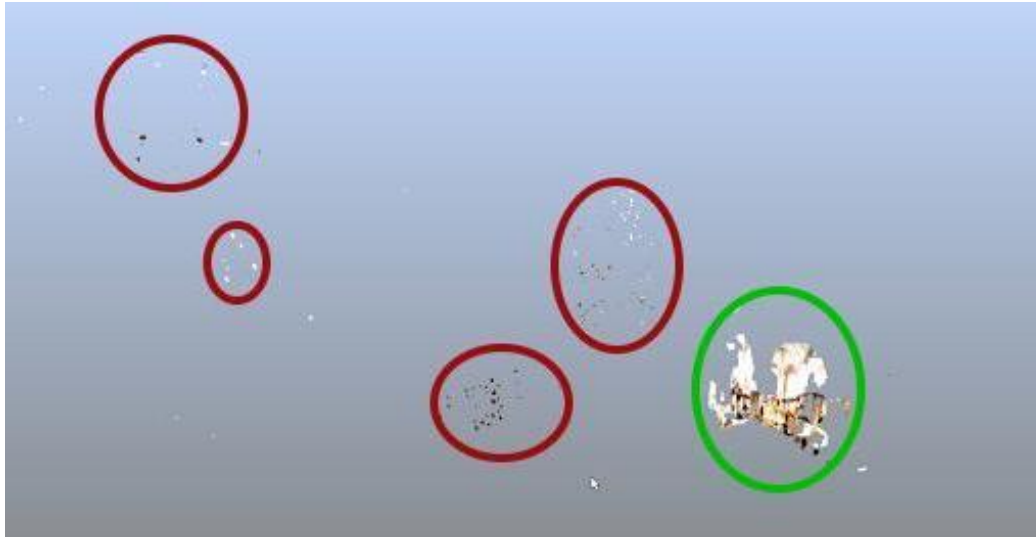


Figure 6. Example of glitches (red regions) very far from scanned object (green region).

Unwanted objects

Sometimes in the point cloud we can find distorted, misplaced or duplicated objects. These glitches are usually caused by movement during the scanning - people, animals or any changes in the scenery. Similar effects may result from reflections. However we may want to erase other not important objects that could not be removed on the set. The Figure 7 depicts a person walking behind the pillar.

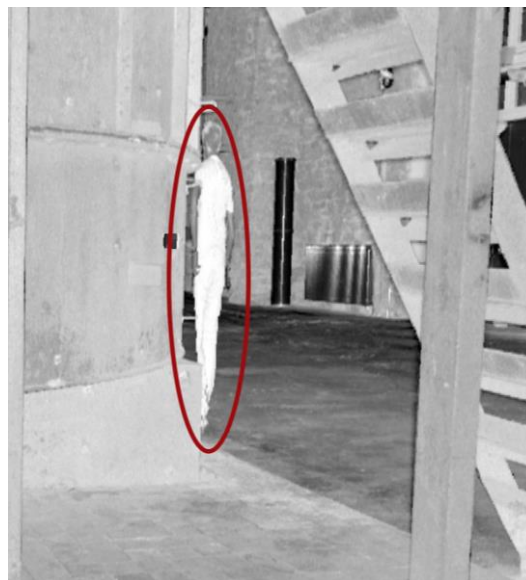


Figure 7. Example of unwanted object

3.5 Exporting

The final stage of creating a point cloud, after completing the entire processing, is the export of data in one of the standard formats. In our case we used e57 format, because it is vendor-independent and widely supported by third-party 3D software. Using well-defined and standard based format is a good practice as it can be easily used in many different applications.

The stitching of cathedral scans took over 30 hours, mainly due to the fact that the scans were placed manually for the most part. Usually, the software automates this process, but it requires placing reference objects in the scanned scene (reference spheres targets, checker board, etc.). We did not

decide to use this in our workflow from an aesthetic point of view, because later removal of such objects from the scans is also difficult and time consuming.

3.6 Good practices

The processing time of point cloud compiled from multiple scans may be long. To decrease that value, remove unnecessary and redundant points. Lowering point density will decrease the time of processing, but may also decrease the quality of renders.

In order to simply visualize the created point clouds in real time you can use PoTree⁶ online web renderer based on JavaScript and WebGL.

4 Rendering

4.1 Introduction

As mentioned earlier, the point cloud is a set of data describing the location of points in three-dimensional space with their assigned colour. The data can be displayed using a special point cloud processing program, but to generate a high-resolution video from the scanned and processed data, a rendering process must be performed.

Due to high density of input points, it is possible to generate video materials in ultra-high resolution with very good quality, e.g. 16K or even more, without losing details and sharpness of the image. There are several applications on the market that enable photorealistic renderings based on point clouds. In our workflow we used two of them to render 8K, 16K, 360° and 3D content: CloudCompare for rendering Cathedral church and Blender for further projects.

4.2 Image format

Prior to setting the parameters and rendering, the most important thing is to determine the resulting image format, as this imposes important limitations.

4.2.1 2D Images

The simplest case is to render a two-dimensional image. The resolution of the resulting image should be decided, but also the angle of the virtual camera lens. Of course, depending on the specific point cloud and the purpose of creating the resultant film, we can experiment with these settings, but in the Cathedral we used the 100 degrees angle, because it gave the best visual results in this interior.

4.2.2 Stereoscopic 3D

To render a stereoscopic image, generate images for the right (R) and left (L) eye separately. Additional parameters that define stereoscopic vision are: base - distance between cameras and convergence -

⁶ Potree web rendering software - <http://potree.org>

angle of their convergence. The place of intersection of both camera axes is often referred to as focal point. Because the Camera in CloudCompare does not have all the features of the 3D Camera (see section 4.3.5) in the case of the Cathedral rendering, we had to use some simplifications: the stereoscopic image is generated using only a shift of the camera position (base). However, the angle of convergence of the cameras' axes was negligible, because in vast spaces the focal point tends to be infinite.

4.2.3 360° format

The 360° image shows a view of the entire sphere around the camera. As the codec uses two-dimensional images for encoding, it is necessary to project the sphere onto a flat image. We use the so-called projection, i.e. the geometric transformation of the image to match the frame dimensions. There are many methods of performing 360° projection, but currently two types of it are commonly used: Equirectangular Projection (ERP) and CubeMap Projection (CMP).

It is possible to convert between different types of sphere projections into a two-dimensional image frame, however, such a conversion is always associated with a loss of image quality.

Equirectangular Projection (ERP)

This projection is also called the "non-projection", or *plate carrée*, since the horizontal coordinate is simply longitude, and the vertical coordinate is simply latitude, with no transformation or scaling applied. The equirectangular projection was used in map creation since it was invented around 100 A.D. by Marinus of Tyre. In an equirectangular panoramic image all verticals remain vertical, and the horizon becomes a straight line across the middle of the image. Coordinates in the image relate linearly to pan and tilt angles in the real world. The poles (Zenith, Nadir) are located at the top and bottom edge and are stretched to the entire width of the image. Areas near the poles get stretched horizontally.⁷



Figure 8. Example of ERP projection

⁷ [PanoTools Wiki - Equirectangular Projection](#)

CubeMap Projection (CMP)

In CMP the images are arranged like the faces of a cube, with each face in the Rectilinear Projection. This cube is viewed from its centre. Four cube faces cover front, right, back and left, one the zenith and one the nadir, each of them having 90°x90° Field of View. In each cube face all straight lines stay straight, hence it is very good for editing.⁸

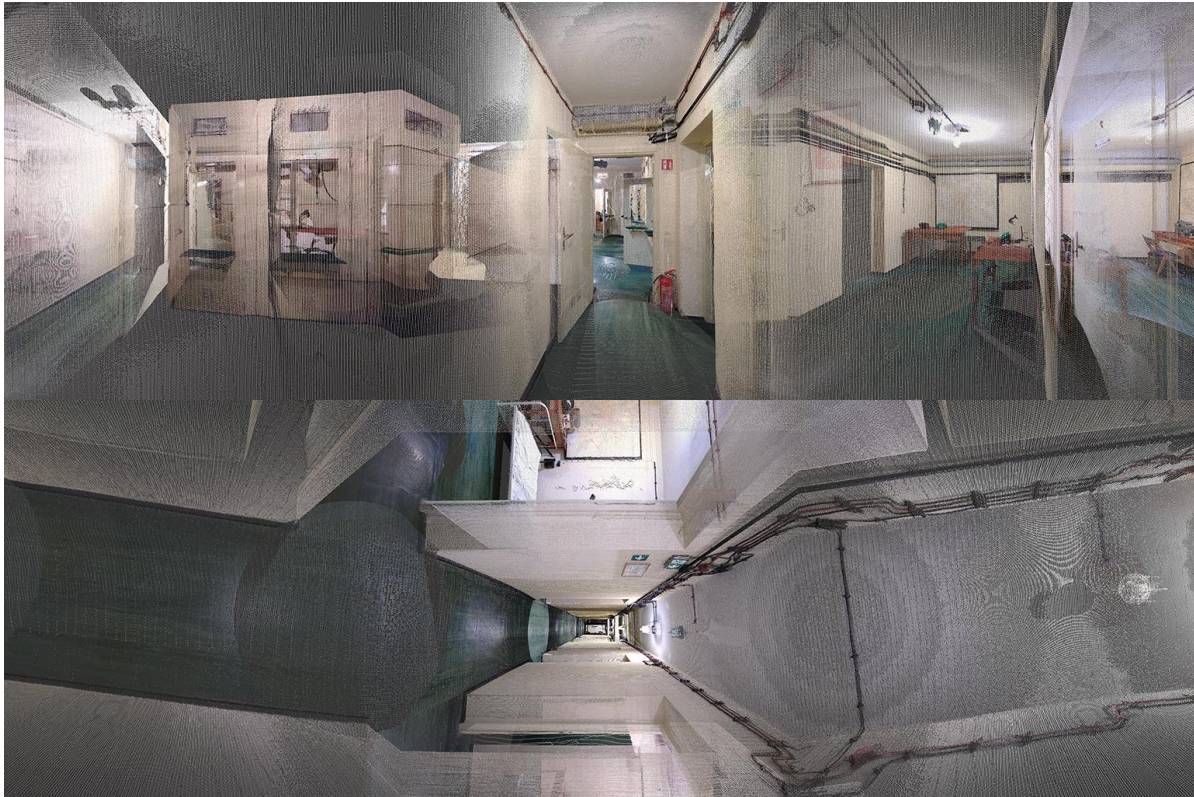


Figure 9. Example of CMP projection

4.3 Content rendering using CloudCompare

4.3.1 CloudCompare Application

The CloudCompare toolkit is an application for viewing, editing and processing huge point clouds stored as an octree structure. It supports many open point cloud formats such as ASCII, LAS, E57, etc. as well as vendor-specific formats (DP, RiegI, Faro). Although the software provides many methods of point cloud processing and output generation of rendered images, the developers did not assume applications for resolutions above 8K. In order to use the full potential of the software in the Immersify project, especially to create high quality and high resolution video, it was necessary to implement changes related to rendering up to 16K resolution and advanced virtual camera tracking using Bézier curves. We also implemented several additional settings and parameters which make the production of 360° and 3D images much easier and efficient. The changes are related to two modules:

- *qAnimation* – a few significant changes were added to this module, i.a. advanced movement interpolation based on Bézier curves and splines with editable parameter; rotation

⁸ [PanoTools Wiki - Cubic Projection](#)

interpolation; configurable settings for camera position and rotation, resolution of image, frame range, frame ratio, watermarks, naming).

- *qWrlAnimation* – added a new plugin for rendering video frames based on path imported in standard .wrl format generated in any third-party applications for path edition (e.g. 3D Studio). The plugin also supports functionality of: watermarking of output frames, selecting camera from WRL files, editing relative position and rotation of the camera and frame resolution beyond 8K.

These additionally implemented options also support generation of 360° and 3D images, by using the same trajectory of the virtual camera we can control two different virtual cameras with defined offset for stereoscopic production. In a similar way, it is possible to control the virtual cameras for generating cube-map images required for 360° content production but in this case we have to play with angle rotation instead of offset.

The work done by PSNC will be contributed to the open source software on GitHub. To use it just clone our forked repository and follow the building instructions from *BUILD.md* file. Our fork is available under the following link: <https://github.com/PSNC-NM/CloudCompare>. All additional libraries are described in *BUILD.md* file.

The entire process of content rendering in CloudCompare software consists of three main steps which results in a sequence of graphic files in a selected format (e.g. PNG). All steps are briefly described in the following sections.

4.3.2 Preparations

In the first step you have to import into CloudCompare a selected point cloud generated with FARO Scene. Before loading the points cloud, you need to configure the main view of CloudCompare and remove gradient background and change view perspective. In order to do it select Display options and uncheck “display gradient background” from Colors and Materials tab, and then set view mode for Viewer-based perspective.

4.3.3 Path definition

The way of visualization and presentation of point clouds on 2D/3D displays depends on the defined path of virtual camera movement in the whole scanned space. Each path consists of a sequence of defined points called “Viewports” between which the camera moves continuously. To create a viewport in CloudCompare simply place a virtual camera at a selected position via mouse or open camera settings from side menu (Figure 10), set camera position, rotation and field of view and press Ctrl+V. CloudCompare *qAnimation* plugin requires minimum three viewports: starting and ending points and at least one inflection point between them, but you can define as many points as required to achieve fluent and smooth visualization. In a specific case, if you want to achieve a straight motion path of a virtual camera, simple duplication of beginning or ending viewport is required.

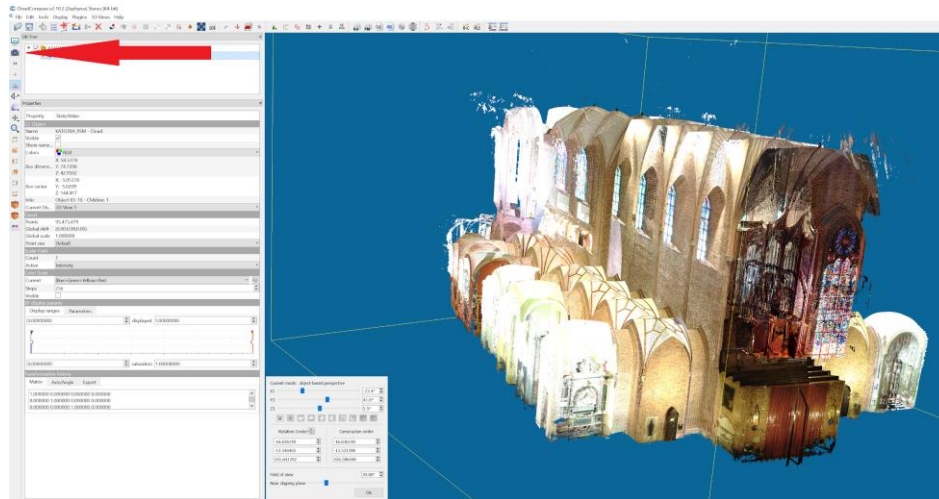


Figure 10. Opening camera settings.

4.3.4 Animation and rendering

Animation in the context of video material generated from a point cloud is, in general, a smooth transition of a virtual camera between defined "Viewports" starting from the starting point through all intermediate points up to the ending point. Thanks to the implementation of extensions to the qAnimation plugin, it is possible to precisely control all the parameters of this movement (e.g. duration, interpolation settings) as well as all the settings related to the quality of the generated video material. Prior to opening the qAnimation plugin window select at least three viewports (start, intermediate, end of animation), then go to Plugins->Animation from top menu and set all the required parameters (Fig.11). When the parameters are set, use the preview button to preview animation and export (Export frames button) it as a sequence of PNG files (common, supported by a wide range of software, lossless, 16-bit image format). In order to prepare 2D content use the front camera view, to prepare renders for 360° content in CMP projection export six square faces of each viewing direction. To render square frames check Ratio 1:1 checkbox and to change rendered face change value of relative rotation, respectively:

- front – X:0 Z:0 Y:0,
- back – X:0 Z:0 Y:180,
- top – X:-90 Z:0 Y:0,
- bottom – X:90 Z:0 Y:0,
- left – X:0 Z:0 Y:-90,
- right – X:0 Z:0 Y:90.

Remember to change the output filename for each face.

List of parameters available in the qAnimation plugin:

- Total duration of animation.
- Duration from each step to the next one.
- Render borders – e.g. to render only the last 50 frames.
- Loop – loops the preview of animation
- Interpolate – smooths the path of animation.
- Angle mode – smooths the rotation of the camera.

- Parametrize spline – value of smoothing.
- Relative position in each axis (e.g. for creating 3D animation).
- Relative rotation in each axis (e.g. for creating cubemap).
- Frame rate of output.
- Bitrate – ignored.
- Super resolution/zoom – ignored if Static size is checked.
- Predefined static sizes can be used with ratio 1:1 – width:width.
- Output files path and name.
- Watermark file path – it should be an overlay in export size, may be empty.

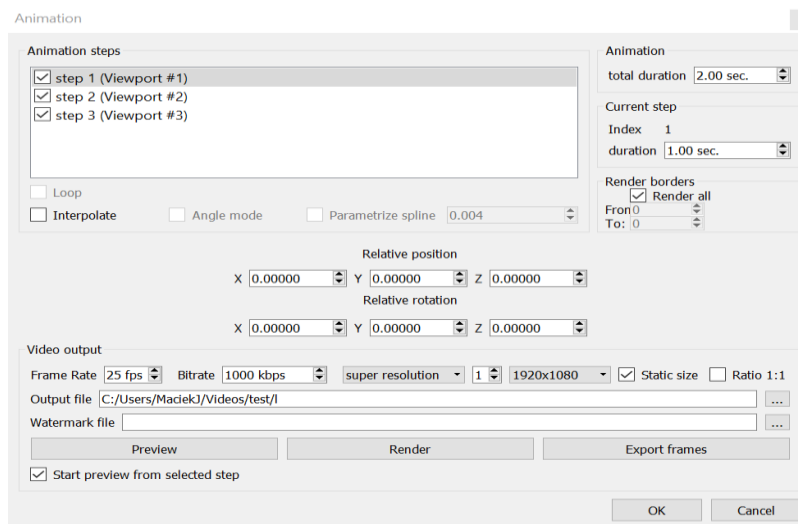


Figure 11. qAnimation Plugin

4.3.5 3D and limitations

During creating workflow for point cloud rendering, we encountered several problems. CloudCompare has quite large limitations. It is primarily a program for editing and previewing point clouds, and not for creating animations with views other than 2D - the space used here is called "2.5D", which means overlapping appropriately calculated layers on a two-dimensional image. Similar concept is used, among others, by Adobe After Effect. CloudCompare does not support natively three-dimensional space, but only simulates the depth of the image. This limitation is irrelevant for two-dimensional images and allows you to efficiently visualize data at high resolutions. Unfortunately, some operations, such as 3D stereo rendering, are very limited, and others are even impossible (e.g. 360° Stereo).

In the case of Stereo 3D, a favourable circumstance was the large volume of the object to be visualized. The long distances between the observer and the elements of the cathedral made it possible to assume a simplification that focal point is aiming at infinity. Thus, the 3D image was generated by generating an image from two cameras with parallel optical axes, which are spaced by a distance close to the base of human eyes – 7 cm. Due to a very unusual description of camera rotation in CloudCompare, the animation has been limited to camera positions.

4.4 Good practices

- The quality of built-in video encoder is quite low, so we do not recommend using it if the high quality is the priority.
- It is recommended to render to a sequence of frames using a lossless format (png, tiff, dpx), and encode them later with specialized tools.

4.5 Rendering using Blender and PointCloudVisualizer

Due to the complexity of CloudCompare, in order to overcome all the limitations for 3D rendering, it would require rewriting almost the entire engine. We were looking for a solution that would allow more flexibility in dealing with the problem of depth of the image. The idea was to base the workflow on software using natively a full three-dimensional space. We found that the Blender and its plugin called Point Cloud Visualizer could make the work. It allows simple editing, elementary transformations and visualizations of point clouds using the PLY file format. Blender code is written in Python, which further simplifies the process of adapting this tool to our needs.

Blender is a program with a set of tools prepared for animation, so working with a virtual camera is easier. The tools of this program allow you to control every aspect of animation. Dope sheet module allows full control over keyframes, and the graph editor allows, very conveniently, to control and change all of the animation parameters.

At present, in our workflow we export a point cloud, which is composed in scanner dedicated software Scene, to ASCII cloud format file (ie. .xyz or .pts). With the help of Cloud Compare described earlier, we need to convert the data to PLY format supported by PVC add-on in Blender. After applying the PCV plugin to any empty object and loading the cloud from the file you can start working on it by hitting the Draw button. PCV allows quite advanced editing capabilities and operations performed on loaded data. However, when generating stereoscopic (3D) images there is a small inconvenience. Because the plugin does not support any of the built-in Blender renderers, we are forced to construct our own 3D rig.

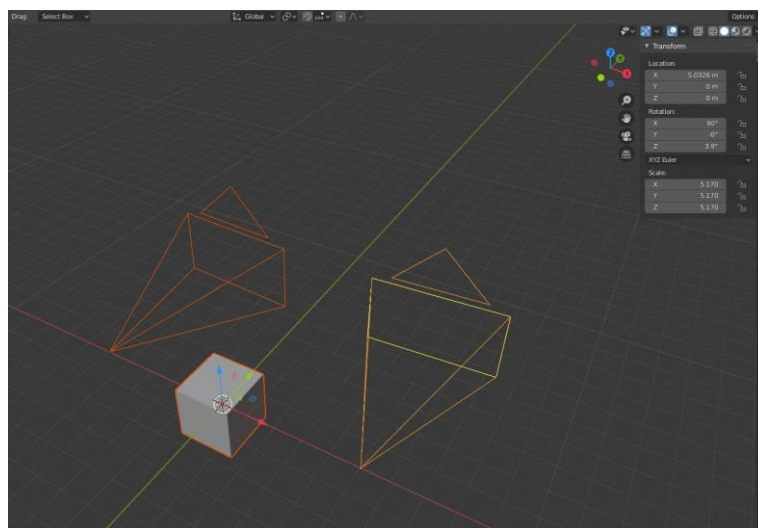


Figure 12. Proposed virtual stereo rig

This comes down to the fact that the two "traditional" Blender cameras are linked to each other, with the distance between them as a stereo base and the angle between their axes as convergence. Frames rendered from individual cameras using After Effect, Fusion or any other graphics software are combined as side-by-side or up-down images for the appropriate eye, depending on the format adopted.

At the moment, workflow is still being tested and modified and for this reason it is difficult to specify its final form. The development of the optimal workflow will be the subject of our further work in the future.

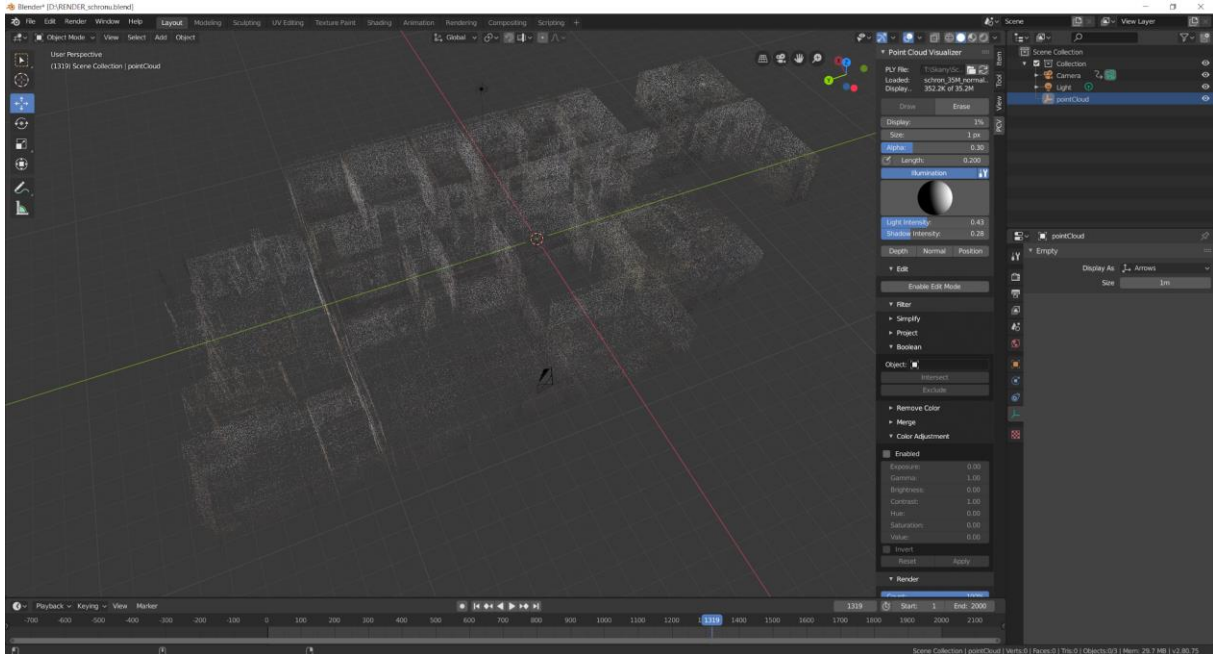


Figure 13. The Blender interface with point cloud imported (1)

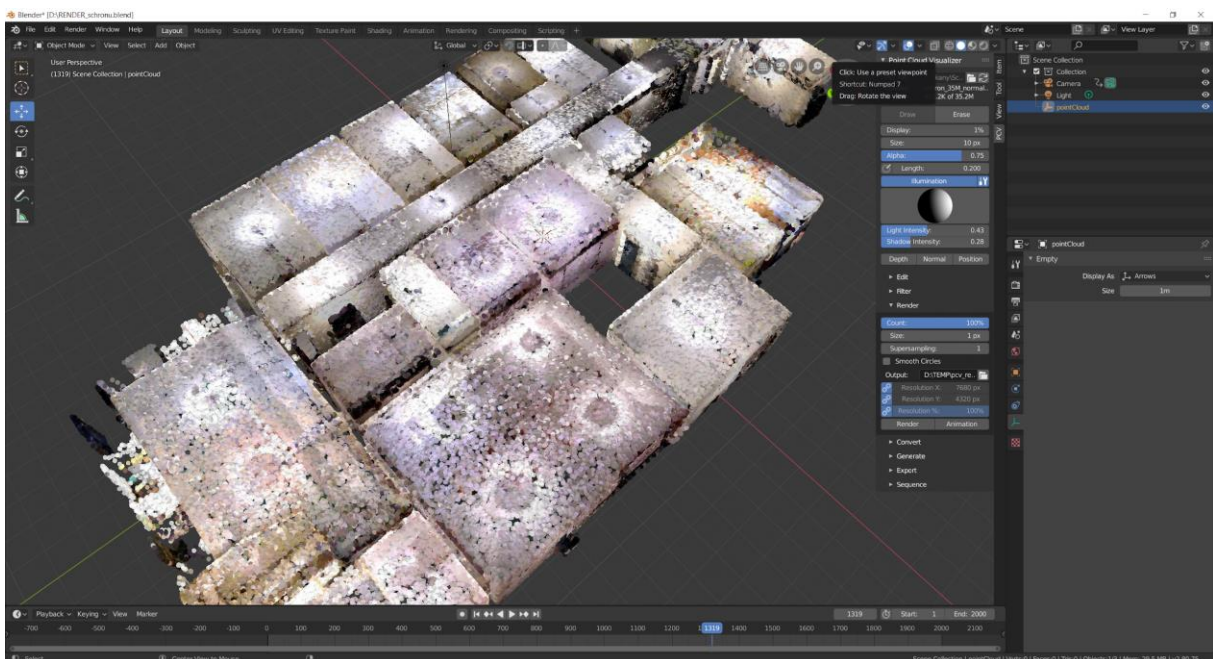


Figure 14. The Blender interface with point cloud imported (2)

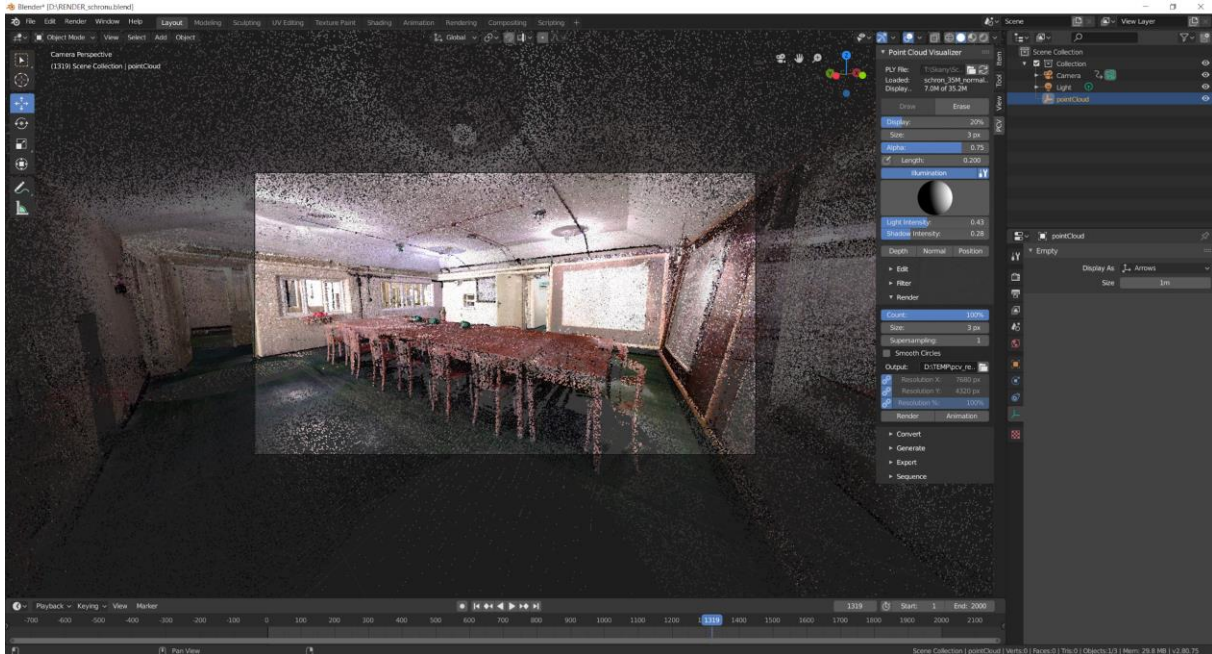


Figure 15. The Blender interface with point cloud imported (3)

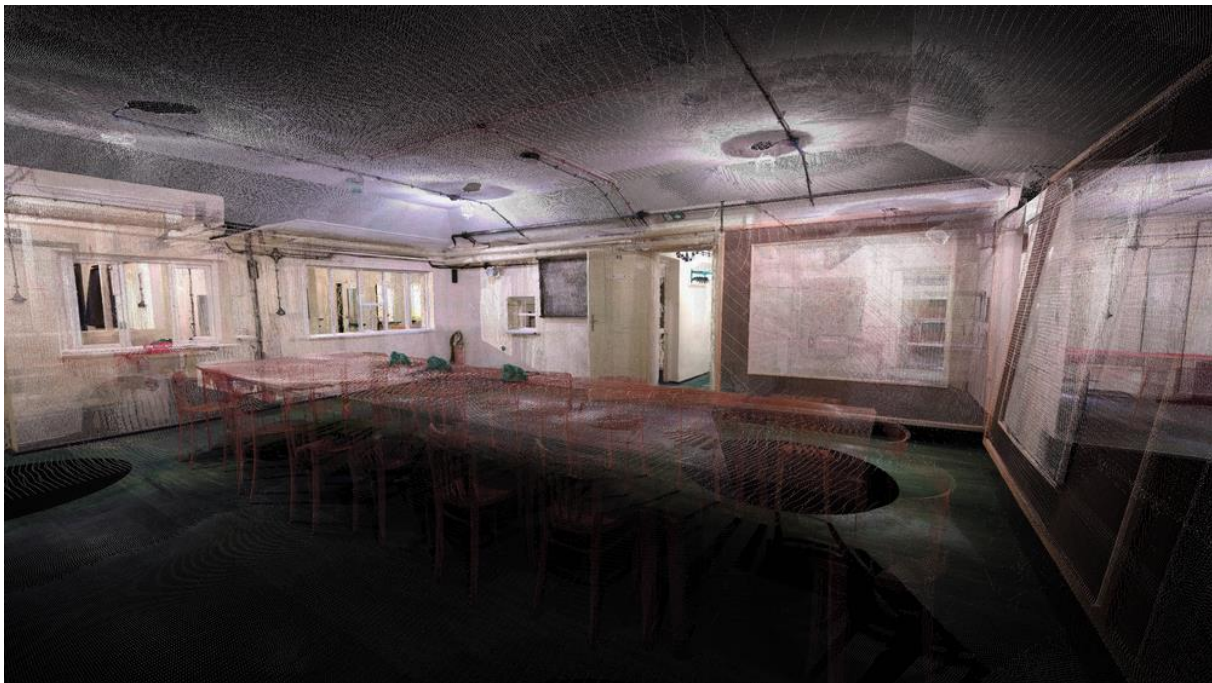


Figure 16. Test renders in Blender

5 Post-production and Encoding

5.1 Image processing

After the rendering stage in CloudCompare, Blender or any other application, we obtain a ready set of single graphic files containing successive frames of the prepared animation. In order to prepare appropriate materials for different types of content (2D, 3D, 360°) and displays (HMD, CAVEs), post-production process should be conducted, including appropriate preparation of graphic formats, colour

correction and the final encoding process allowing for the display of materials with the use of standard video players. Real-time rendering and displaying point cloud animation at high resolution and bitrate is still a challenge even for modern high-spec workstations. So, the best way of optimization is to encode the content and generate standard video files.

5.2 Encoding

The way and scope of preparation generated images for the encoding depends on the type of content, type of visualization installation and selected encoder we are going to use. In this guideline we assume the use of Spin Digital HEVC encoder. High Efficiency Video Coding (HEVC, or H.265) is a widespread video standard developed by MPEG, which is able to compress video in a very efficient way preserving very high quality, in comparison to H.264, it requires about 50% less bitrate at the same quality level.

Below we present a detailed description of the preparation of the three different types of content (typical 2D animation, stereoscopic content and 360°) using a command line version of SD encoder based on ffmpeg (spinffmpeg) syntax. In general, the point clouds images in UHD resolutions may require the bit rate from 200Mbps to even 2Gbps. The higher bitrate, the better quality, but more CPU/GPU power is required to play the video.

5.2.1 2D content

Typical 2D content is the simplest case because after the rendering stage we have ready to encode a sequence of frames (graphics files) and the only thing we have to do is use a proper encoder command that generates the final video file according to the given parameters. The command below will encode the content in 8K resolution with the constant bit rate at 400Mbit/s:

```
spinffmpeg -framerate 60 -i INPUT_FILES_SEQUENCE -vf
"spinfilter=\ 'a:w=7680:bd=10:cm=yuv:cf=420:cs=bt709l:tf=sdr\ ' -c:v
spinenc -b:v 400000k -o "--rc cbr --fif 1 --ip 64 -g 4" OUTPUT.mp4
```

5.2.2 Stereoscopic content

In case of 3D stereoscopic it is required to generate a sequence of frames in top-bottom configuration which is supported by SpinDigital encoder and player. Views for left and right eye generated in separated files as an output from the rendering process have to be combined into a single image. Encoding procedure and command is similar to the one being used in case of encoding 2D content, but in the first step a top-bottom single image is created and also the target bit rate of the output video stream is increased.

```
spinffmpeg -framerate 60 -i LEFT_EYE_INPUT_FILES_SEQUENCE -framerate 60
-i RIGHT_EYE_INPUT_FILES_SEQUENCE -filter_complex
"[0:0][1:0]vstack[v];[v]spinfilter=\ 'a:cf=422:bd=10\ ' -c:v spinenc -b:v
600000k -o "--rc cbr --fif 1 --ip 64 -g 4" OUTPUT.mp4
```

5.2.3 360° content

In case of 360° content we use traditional Cubemap projection format (CMP) proposed by the Joint Video Exploration Team (JVET) of ITU-T VCEG (Q6/16) and ISO/IEC MPEG (JTC 1/SC 29/WG 11) in document [JVET-H1004](#) section 2.2. This format is natively supported by the SpinDigital encoder and player. The CMP 3x2 packing arrangement, with six square faces is shown in Figure 17. The faces in the bottom row are rotated by 90 degrees.

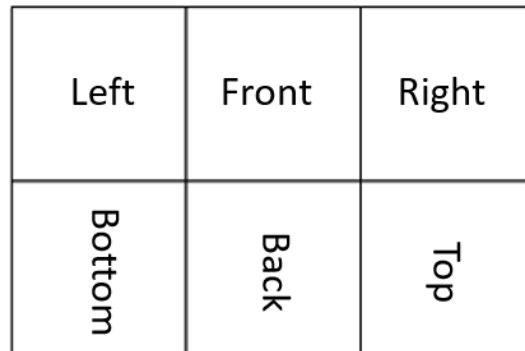


Figure 17. CMP 3x2 face packing

An example of real CMP 3x2 image from the cathedral laser scans is shown in Figure 18. For the stitching of the six square images (faces) into one cubemap image we recommend to use e.g. `imagemagick` software or `ffmpeg` built-in filters (`hstack`, `vstack` or `xstack`).

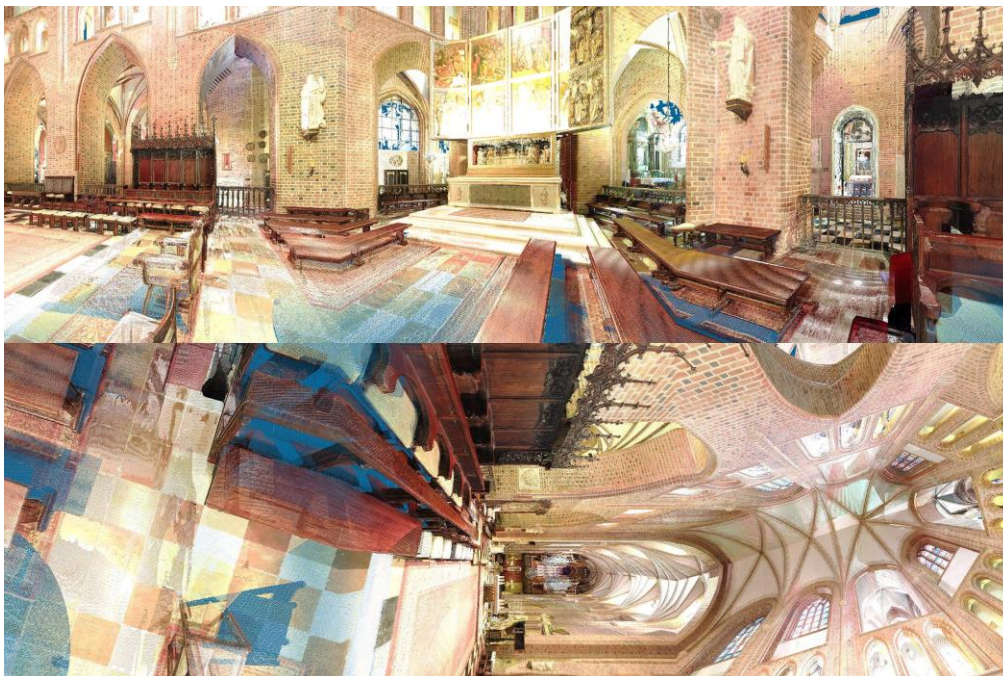


Figure 18. Real frame containing CMP 3x2 with Cathedral church

The below command will encode the 3x2 Cubemap files composed of six 4Kx4K viewports (total resolution: 11520px horizontally) with the constant bit rate at 800Mbit/s:

<pre>spinffmpeg -framerate 60 -i INPUT_SEQUENCE_IN_CMP_FORMAT -vf "spinfilter= \ 'a:w=11520:bd=10:cm=yuv:cf=420:cs=bt709l:tf=sdr\ ' " -c:v spinenc -b:v 800000k -o "--rc cbr --fif 1 --geom cmp --ip 64 -g 4" OUTPUT.mp4</pre>	
<pre>spinfilter=\ 'a:bd=10:cm=yuv:cf=420:cs=bt709l:tf=sdr:geom=cmp\ '</pre>	<ul style="list-style-type: none"> ● bd=10 – sets color depth to 10 bits, ● cm=yuv – Overwrite input color model, ● cf=420 – Overwrite input chroma subsampling ● cs=bt709 – Overwrite input color space. ● tf=sdr – Overwrite input transfer functions to sdr, ● geom=cmp – enable quality enhancements in 360° video.
-c:v spinenc	Target codec of a video stream.
-b:v 800000k	Target bit rate of a video stream.
-o "--rc cbr --geom cmp --fif 1 --ip 64 -g 4"	<ul style="list-style-type: none"> ● rc cbr – Rate control, Constant Bitrate ● fif 1 – frames-in-flight produces slower encoding than the default one (16 frames-in-flight for VBR, 4 frames-in-flight for CBR) but produces more stable bitrate and higher quality video, ● geom cmp – enable quality enhancements in 360° video, ● ip 64 – Number of pictures between IRAP pictures. Automatically rounded to the nearest multiple of the Group of Pictures (GOP) size ● g 4 – Maximum GOP(Group of Pictures) 4 frames.