

# AN AUTOMATED PROCESS TO DETECT EDGES IN UNORGANIZED POINT CLOUDS

A. Mitropoulou, A. Georgopoulos\*  
Laboratory of Photogrammetry, National Technical University of Athens, Athens, Greece  
mitr\_cath@hotmail.com, drag@central.ntua.gr

Commission II, WG II/8

**KEY WORDS:** Point clouds, plane detection, RANSAC, edge detection

## ABSTRACT:

This paper presents an automated and effective method for detecting planes and their intersections as edges in unorganized point clouds. The edges are subsequently extracted as vectors to a CAD environment. The software was developed within the Microsoft Visual Studio and the open source Point Cloud Library (PCL, <http://pointclouds.org/>) was used. The Point Cloud Library is a standalone, large scale, open project for 2D/3D image and point cloud processing. The code was written in C++. For the detection of the planes in the point cloud the RANSAC algorithm was employed. Subsequently, and according to the standard theory of Analytic Geometry the edges were determined as the intersections of these planes with each other. A straight line in 3D space is defined by one of its points, which was determined with the Lagrangian Multipliers method and a parallel vector, which was determined with the help of the cross product of two vectors on space. Finally, the algorithm and the results of the implementation of the process with real data were evaluated by performing various checks, mainly aiming to determine the accuracy of the edge detection.

## 1. INTRODUCTION

Nowadays, with the advancement and the popularity of Image Based Modelling, but also with the increasing use of terrestrial laser scanning an infinite number of point clouds are acquired, especially for the geometric documentation of cultural heritage artifacts and monuments (Kersten et al. 2009, Tryfona & Georgopoulos, 2016). Their processing and correct exploitation to contribute to the final documentation products need some attention, as it poses a severe bottleneck in the workflow. On the other hand, in the cultural heritage domain and especially for the restoration studies, 2D vector drawings are still required by the conservation experts. Consequently, there is an increasing demand for exploiting the acquired point clouds to produce 2D vector drawings.

Certain researchers (Rodríguez Miranda et al. 2008, Canciani et al. 2013, Azariadis & Sapidis 2005) have tried to produce vector shapes, i.e. mainly linear features, directly from the point clouds either manually or semi-automatic. However, this procedure is time consuming and extremely strenuous. Besides, it requires powerful computers to load, if at all, the point clouds with their millions of points. Consequently, the solution lies in the automation of vector detection directly in the point cloud (Briese & Pfeifer, 2008).

Several efforts have been published so far for that task with varying degrees of success (Huang et al. 2003, Vosselmann et al. 2003, Lu et al. 2008, Yu et al. 2015). Their main purpose has been to detect straight edges in the point clouds, which is one of the most important issues in the area of image processing and computer vision, and which has been studied for years. In general, their detection is carried out as the boundary between areas of different luminance or texture. However, the methods which have been used to detect edges in images cannot be applied to unorganized point clouds because the latter have a different structure from the images (<https://stackoverflow.com/>). An image is a matrix, while a point cloud is an irregularly

distributed set of data. This differentiation in their structure, resulting in the non-usability of methods applied to the images, has led to the detection and extraction of edges from unorganized point clouds as a new research challenge (Briese, C., Pfeifer, N., 2008).

An edge detection technique from point clouds was developed by Bazazian et al. (2015). This technique has been able to extract edges by analyzing the covariance matrix eigen values calculated by the nearest neighbor's algorithm of each point. A further edge extraction algorithm is proposed by Hackel et al. (2016). The authors point out the importance of detecting edges along which their orientation changes. They use a two-step method. Firstly, a score with a binary classifier is given at each point in the point cloud, using features exported from its neighboring points. In the next step, another binary sorting is performed to select the optimal set of points belonging to the ultimately extracted edge.

With the goal of fully or partially automated linear vector extraction from unorganized point clouds, the current study aims to locate planes in order to determine their intersections and finally extract them as edges, i.e. vector lines, of the depicted object to which these planes belong.

The rest of the paper is organized as follows: Section 2 explains the point clouds, Section 3 shows the proposed methodology, Section 4 reveals the results and finally, Section 5 exposes the conclusions.

## 2. POINT CLOUDS

Common terrestrial laser scanning systems in general produce unorganised point clouds. An organised point cloud is organised as a 2D array of points with the same properties as if the points were obtained from a projective camera, like the Kinect, DepthSense or SwissRanger. In both organised and unorganized

point clouds, all of X Y and Z are provided for each point, but the memory layout of organised point clouds is that of a 2D array. The memory layout of the points then is closely related to the spatial layout as represented by these XYZ values. Algorithms that work on unorganised point clouds will generally work on organised point clouds (<https://stackoverflow.com/>). Organized point clouds resemble the structure of an image. In this case, the first record consists of the point in the first row and in the first column, the second record is the point in the first row and in the second column, the third entry is the point in the first row and in the third column and so on. Figure 1 gives an example of an organized point cloud and its file.

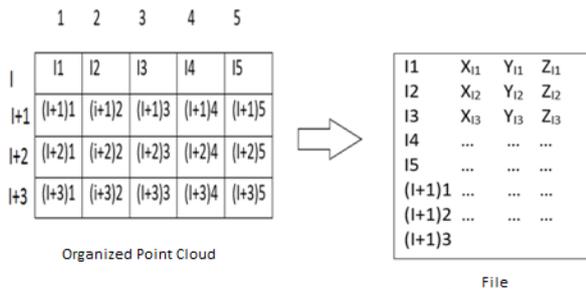


Figure 1. Structure of an organized point cloud

By contrast, in an unorganized point cloud the records in its file are entered by column, i.e. the first entry is the point in the first row and the first column, the second entry is the point in the first column and in the second row, the third entry is the point in the first column in the third row and so on. Figure 2 depicts an example of an unorganized point cloud and its file.

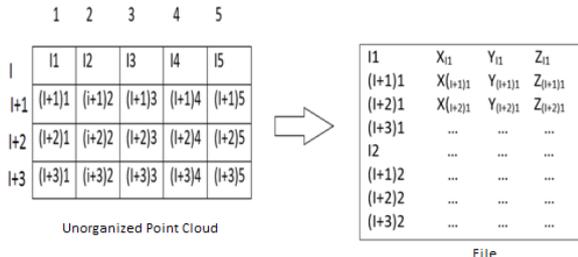


Figure 2. Structure of an unorganized point cloud

The unorganized point cloud feature can easily be seen in a point cloud data (.pcd) file, specifically in the HEIGHT field of the file (Figure 3), where in these cases it is equal to 1, i.e. the records form a continuous sequence and not a matrix-like one.

```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z
SIZE 4 4 4
TYPE F F F
COUNT 1 1 1
WIDTH 232463
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 232463
DATA ascii
82.945175 100.52753 1.7449874
82.96431 100.53015 1.7428735
82.974838 100.5444 1.7439721
82.983292 100.47661 1.7441075
82.993019 100.5485 1.7422022
83.002861 100.47902 1.7433289
83.012909 100.49424 1.7432919
83.032753 100.4965 1.7422633
```

Figure 3. A point cloud data file (.pcd) for unorganized point cloud data

For this reason, Analytic Geometry was selected as the methodological tool to detect the required edges as plane sections. This, of course, presupposes the initial knowledge of the plane equations that were identified in the point cloud. For that purpose, the RANSAC algorithm (Fischler & Bolles, 1981) was chosen as the ideal technique.

### 3. PRINCIPLE OF PROPOSED METHOD

The most important feature of the cloud with a key role in selecting the method for developing the program was the fact that the cloud is unorganized, i.e. the cloud's file is unstructured.

The proposed method for the determination of edges as plane sections comprises of three steps:

- i. Detect planes in the given point cloud
- ii. Determine the plane intersections
- iii. Extract the intersections as vector edges, ready to be inserted in a CAD environment

In order to determine the mathematical model of the planes, the four parameters of the plane's equation (A, B, C and D) should be calculated. It is known that three non-collinear points define a plane. The RANSAC algorithm is implemented to determine the plane in the point cloud as follows: i) random selection of three points and ii) estimation of the plane's parameters. Then, based on a predetermined threshold, the number of additional points that can be included in the calculated plane is noted. Thus, the process is repeated for all possible triplets of the point cloud. Finally, the plane which has the most inliers in its solution is the most probable one. From this plane, the optimal mathematical plane equation is then calculated.

The edges, i.e. straight lines in 3D space, are detected as sections of non-parallel planes. A straight line in 3D space is defined by one point and the vector parallel to the straight line. The point of the straight line in 3D space was calculated using the Lagrange Multipliers Method with two constraints, so that the point is confined within the range of the point cloud coordinates (1<sup>st</sup> constraint) and at the shortest distance from the beginning of the axes and not to be any point of the straight line (2<sup>nd</sup> constraint). The vector parallel to the straight line was calculated with the help of the cross product of the planes, to which is identical.

The last step is the extraction of the edge points, so that they are immediately usable in a CAD environment. This process was based on the Vector Projection algorithm (Lu, Chen & Jianguo, Cai, 2003). More specifically, the coordinates of the projected points, i.e. the points of the edge, are calculated by projecting onto the edge of a vector which starts at a point on the edge of the two planes and ends at a point in the point cloud.

The flowchart of this procedure is presented in Figure 4.

### 4. IMPLEMENTATION AND RESULTS

In order to establish the effectiveness of the procedure, various tests were performed and edges belonging to planes with different characteristics were detected. The data used to detect the edges came from a Leica ScanStation 2 laser scanner, with an uncertainty of ±5 to 6 mm compared to the real value of the measurement. Scanning was performed at an average density of 4 mm.

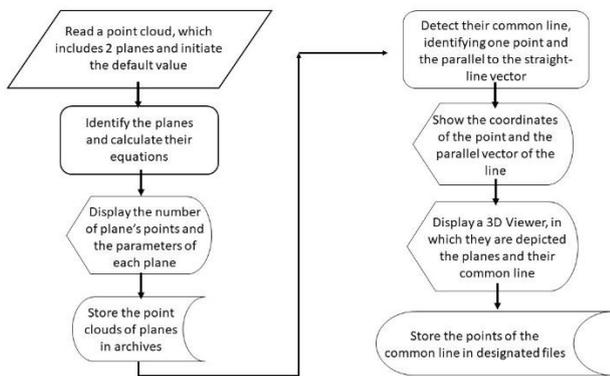


Figure 4. The flowchart of the developed software

The first test was carried out in the point cloud of the planes depicted in red in figure 5. The cloud comprised of 76.175 points and the threshold value for the implementation of the RANSAC algorithm was set to 1.2 cm so that the surfaces approach the form of a mathematical surface and in particular that of a plane within that threshold. In this way, it is easier for the algorithm to find the optimal mathematical model of the surface.

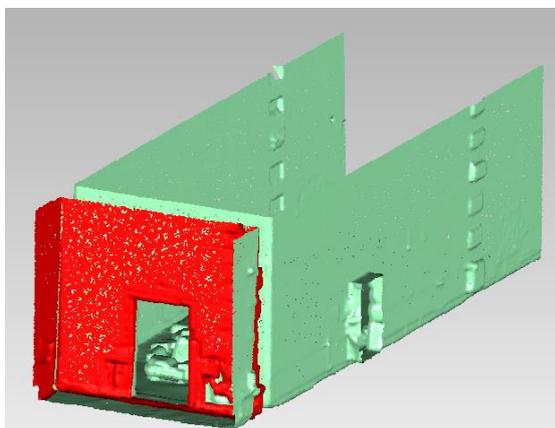


Figure 5. Displaying planes in red to apply the edge detection process.

The results for that point cloud are shown in the image below. The number of inliers calculated for the first plane is 13.738 and for the second is 46.926. Furthermore, the four parameters for each of the two planes, as well as the descriptive elements of the edge, have been calculated. A last piece of information is the number of edge points belonging to the first and second planes.

```

C:\plane_intersection\build\Debug\plane_intersection.exe
PointCloud representing the planar component: 13738 data points.
PointCloud representing the planar component: 46926 data points.
The 'Distance threshold' is: 0.04 Feet

The equation of a plane is ax+by+cz=d and plane's coefficients are a,b,c and d. Therefore:
The model coefficients_1 for the first plane are: a1=0.223678, b1=0.974603, c1=-0.000172545, d1=-122.911
The model coefficients_2 for the second plane are: a2=0.975064, b2=-0.221863, c2=0.00421217, d2=-45.6719

The line of intersection is represented as a point on the line p=(X,Y,Z) and a direction vector n=(Nx,Ny,Nz)
Point: X= 71.7868 m
      Y= 190.632 m
      Z= 0.178233 m
Direction: Nx= 0.00406716
          Ny= -0.00111041
          Nz= -0.999989

The number of points of the first plane which are located on the intersection line, is: 69
The number of points of the second plane which are located on the intersection line, is: 235
    
```

Figure 6. Imaging Edge Detection Results on 2 planes

The results are visualized in figure 7.

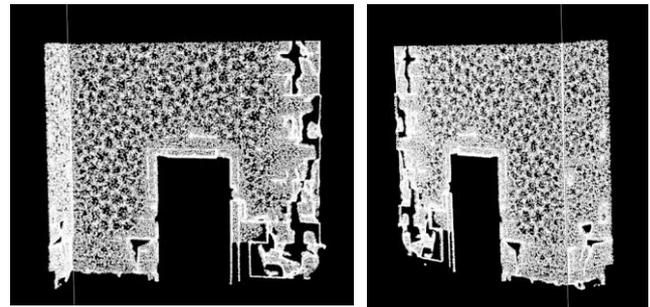


Figure 7. Displaying planes and their edge from two different views

Another important test carried out is to decimate the initial point cloud to 50% and then to 25% of the points. Such a test was considered necessary, as point cloud files are usually very large and consequently the demand in computing resources is extremely high. Hence, very often, they must be decimated, and the robustness of the algorithm was checked in such cases.

Comparing the results, there are no significant differences and therefore no significant deviation between the three edges is expected, namely those calculated with 100%, 50% and 25% of the point cloud. The following table 1 summarizes the results with the number of layer data inliers, parallel vectors in the straight lines, and the number of edge points after the three decimations of the point cloud.

	1st Plane		2nd Plane	
	inliers=	13738	inliers=	46926
Executing Algorithm with 100% of point cloud.	Parallel vector in edge			
	0,004	-0,001	-0,999	
Number of edge points:	69		235	
	69		235	
	1st Plane		2nd Plane	
	inliers=	6972	inliers=	24368
Executing Algorithm with 50% of point cloud.	Parallel vector in edge			
	0,004	-0,001	-0,999	
Number of edge points:	35		122	
	35		122	
	1st Plane		2nd Plane	
	inliers=	3519	inliers=	12512
Executing Algorithm with 25% of point cloud.	Parallel vector in edge			
	0,004	-0,002	-0,999	
Number of edge points:	18		63	
	18		63	

Table 1. Plane detection parameters for various levels of point cloud decimation

These results are visualized in Figure8, showing in purple the points of the edge that have been calculated with 100% of the point cloud, in magenta the points of the edge that have been calculated with 50% of the cloud and black color the points of the edge that have been calculated with 25% of the cloud.

In the second test carried out, one of the two planes is larger than that of the first test. The cloud consists of 327.351 points and the threshold value for the RANSAC algorithm is again 1.2 cm. Figure 9 depicts the planes for which the intersection edge is to be detected.

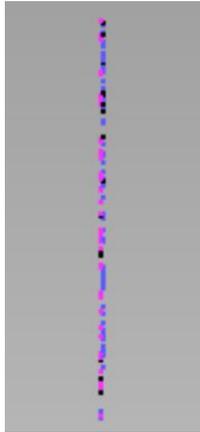


Figure 8. Illustration of the points of the three edges

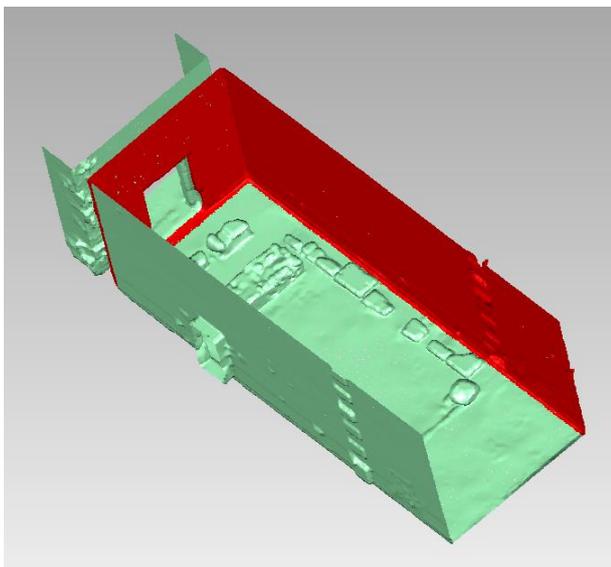


Figure 9. Displaying in red the two planes to apply the edge detection process

After completion of the procedure, a deviation of the edge from the two planes was observed. In particular, the deviation appeared from the middle of the edge with its highest value being at its upper end. The highest deviation value was 4 cm. The results are presented in figure 10.

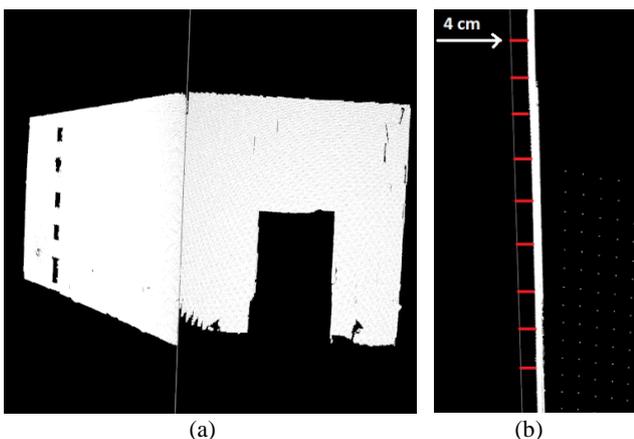


Figure 10. a) Plane and edge imaging b) Edge deviation display

The value of the deviation of the edge rises doubts concerning the accuracy and reliability of the edge detection process. As the theoretical background of the process was thoroughly investigated and it was verified that the procedure followed was correct and appropriate for initially locating the planes and then the edges.

The reason for this deviation was the fact that these surfaces are surfaces in the real world that simply approach the mathematical model of planes. These results in, the RANSAC algorithm computing the optimal plane using inliers data that along the surface do not approach the plane in the same way but are counted for finding the optimal plane. Consequently, the optimum plane deviates from the point cloud and this becomes apparent only at the intersection of the two planes through the deviation from the inliers of the planes. This is most common in larger surface areas. The problem was encountered, giving the program a smaller point cloud from a limited portion of the surface and thus reduced in number of points. Specifically, the points that are close to the desired edge are selected to describe better the surface near the edge through the plane.

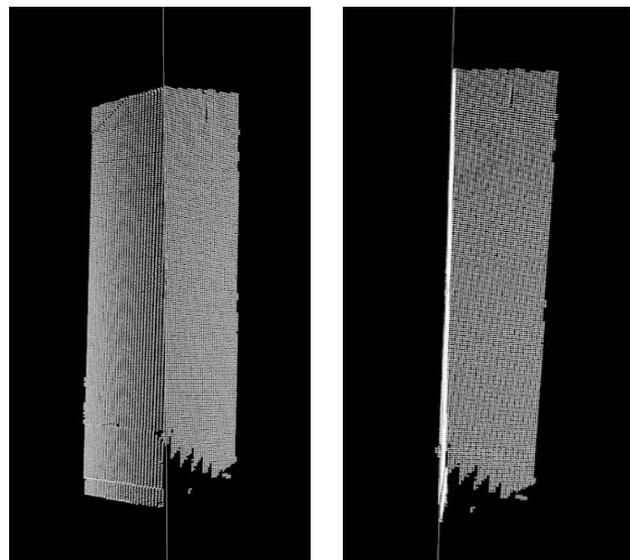


Figure 11. a) Planes and b) edge imaging in two views

To find that this edge is the optimal that can be calculated, the point cloud is even more limited. After this procedure, no edge deviation was observed.

Subsequently, all the major edges of the cloud were detected which was originally selected for testing, with the result being very encouraging for the quality of the edge effects.

## 5. DISCUSSION AND CONCLUDING REMARKS

The evaluations carried out focus mainly on the accuracy of the edge detection and determination. In addition, the extent to which the user can influence this detection has been investigated.

Two important factors influencing the quality of the extracted edges and which can only be determined in advance, are the accuracy and the density with which the point cloud has been acquired in the first place. They directly depend on the equipment used for the measurements as well on as the scanning step. It is obvious that the more accurate the initial point cloud

the better the quality of the extracted edge. Point cloud density affects the number of points defining the planes and, consequently, the extracted edge.

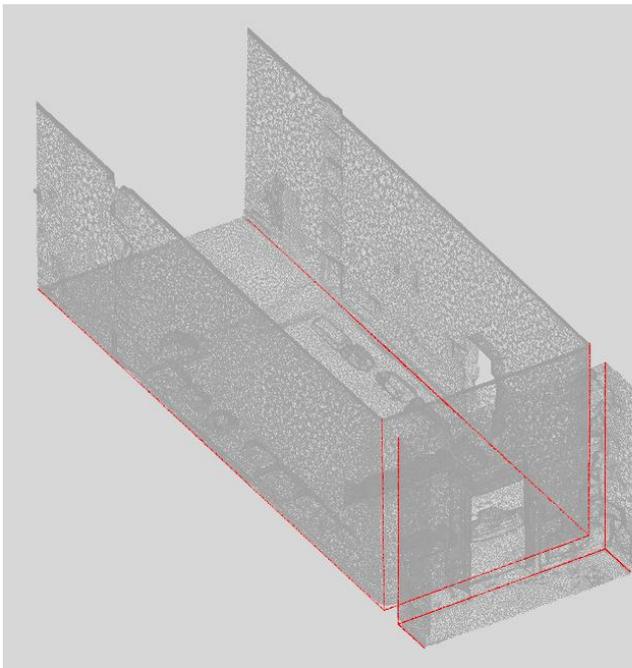


Figure 12. Illustration of the point cloud and edges.

Another factor, which greatly affects the accuracy with which the edges are detected, is the implementation and the specific parameters of the RANSAC algorithm. The user can intervene here to improve its results firstly by properly interpreting the surfaces and their characteristics detected. In addition, through the predetermined threshold, the implementation of the algorithm enables to illustrate this correct interpretation of the characteristics of the surfaces in the results, i.e. to correctly identify the inliers from which the optimal mathematical model of the planes will be calculated. In addition, and in cases of the edge deviations from the determined planes, as the ones observed in the examples presented via the 3D viewer of the program, suitable tests may be performed by each time appropriately limiting the surfaces close to the edge in order to remedy these deviations.

As already described, the process is completed when the differences of the parallel vector of the edge are not significant and thus the most probable best edge is detected. In the presented tests, the results of the edge detection process were very satisfactory given the uncertainty of the scanner measurements and the noise present. The edges were compared to the corresponding edges determined by conventional geometric documentation carried out in the framework of an ongoing project for that purpose.

The tests were performed in part of the point cloud of the interior of the Temple of Hephaestus (Figure 13) in the Ancient Agora of Athens, which was produced for a recent research project. All edges were detected with the proposed method.



Figure 13. The Temple of Hephaestus in the Ancient Agora of Athens

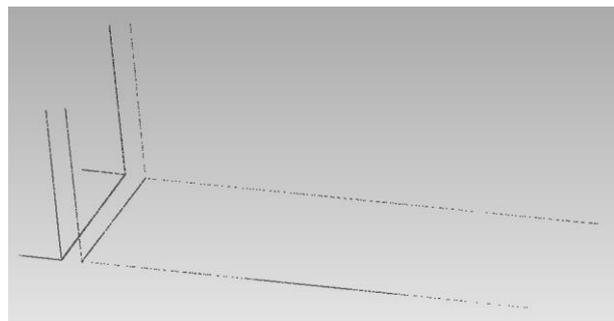


Figure 14. The extracted edges from the point cloud of the interior of the Temple

In order to evaluate the metric precision of the proposed method and the software developed, the extracted edges were compared with the corresponding vector drawings of the geometric documentation of the Temple, which was carried out at a scale of 1:50. Firstly the horizontal edges were put against the corresponding lines of the horizontal section of the Temple (Figure 15), which was measured at 1.50m above the interior floor.

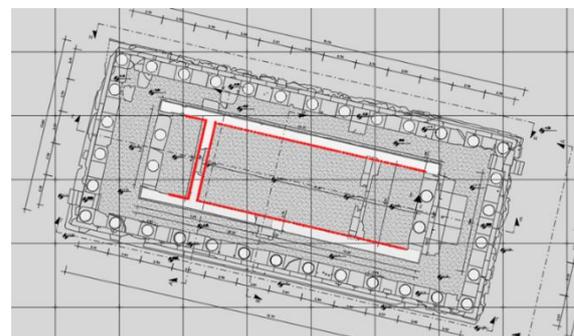


Figure 15. Depiction of the horizontal edges compared to the drawing of the horizontal section of the Temple

The horizontal edges were extracted from suitable parts of the point cloud, i.e. part of the vertical walls approximately 30cm wide at a height of 1.30-1.60m and of the floor, approximately 50 cm wide. The mean differences of the edge pairs were in the order of 5mm (Figure 16).

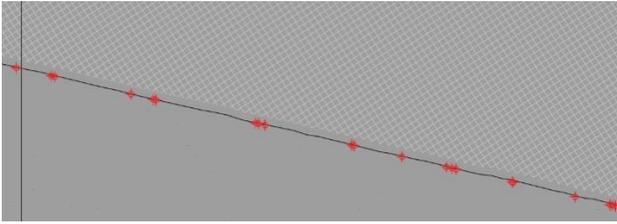


Figure 16: Deviation of the points of the horizontal section from the extracted horizontal edge

These results are quite satisfactory given that the initial uncertainty if the points acquired with the Terrestrial Laser Scanner, i.e. the pulse-based Leica ScanStation 2, is around 5-6 mm according to the manufacturer. In addition, the horizontal edges were detected on the floor level and not at the height of the point cloud section.

The vertical edges extracted were compared against the whole point cloud of the interior. This was performed in the viewer of the software by rotating and observing the deviations from various angles (Figure 17).

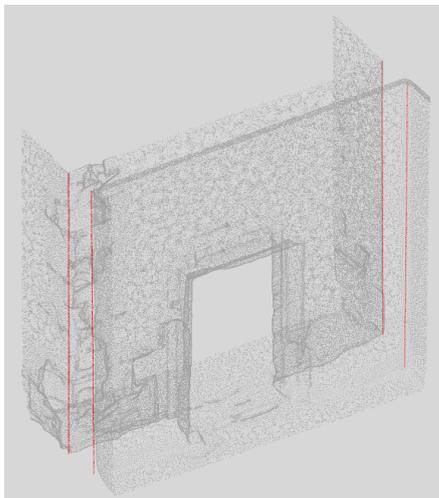


Figure 17: Depiction of the vertical edges extracted against the point cloud

In figure 18 all extracted edges are inserted in the point cloud and the result is very encouraging

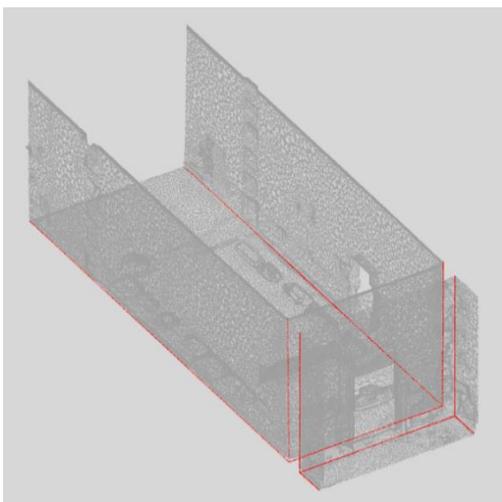


Figure 18: Depiction of the initial point cloud with all edges extracted

After completion of the edge extraction procedure, the available output files are:

1. The files with all points, i.e. the inlier data, which were used to determine the parameters of the two planes. Available in .pcd, .ply and .obj format.
2. The values of the four parameters of the planes, i.e. a, b, c and d.
3. The parameters defining the line in 3D space, i.e. the vector parallel to the line and one of its points, which is closest to the axes origin of the cartesian system
4. The files containing all points on the edges from both planes. Available formats are .pcd, .ply and .obj

A further parameter tested for its influence on the process described, was the number of points in the point cloud. It was tested by reducing the points of the initial point cloud and running the algorithm with the same parameters. It should be noted that this cannot form a general rule as it refers to the particular point cloud. Of course, the fact that this action was not an obstacle to the quality of the outcome, should be a positive element of the process.

Finally, the importance of converting files to point cloud data (.pcd) should be stressed. With this conversion, the organized or unorganized point cloud's structure becomes known, which is very important for the choice of the application method, since some algorithms have as a basic prerequisite the organized structure of the point cloud for their execution.

## REFERENCES

- Alehdaghi, M., Esfahani, M. A. and Harati, A., 2015. Parallel RANSAC: Speeding up plane extraction in RGBD image sequences using GPU. *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*. doi:10.1109/iccke.2015.7365845
- Azariadis, P. N. and Sapidis, N. S., 2005. Drawing curves onto a cloud of points for point-based modelling. *Computer-Aided Design*, Elsevier Publishers, Volume 37, Issue 1, January 2005, Pages 109-122, <https://doi.org/10.1016/j.cad.2004.05.004>
- Bazazian D., Casas J. and Ruiz-Hidalgo R., 2015, Fast and Robust Edge Extraction in Unorganized Point Clouds, In *Digital Image Computing: Techniques and Applications (DICTA)*, Int. Conference IEEE, (<https://imatge.upc.edu/web/sites/default/files/pub/cBazazian15.pdf>, last visit: 10/11/2016).
- Briese, C. and Pfeifer, N. 2008. Line based reconstruction from terrestrial laser scanning data. *Journal of Applied Geodesy*, 2(2), 85-95 January 2008 DOI: 10.1515/JAG.2008.010
- Briese, C. and Pfeifer, N., 2008. Towards automatic feature line modelling from terrestrial laser scanner data. *International Archives of Photogrammetry & Remote Sensing and Spatial Information Science*, Volume XXXVII Part B5 pp. 463-468.
- Canciani, M., Falcolini, C., Saccone, M. and Spadafora, G., 2013. From Point Clouds to Architectural Models: Algorithms for Shape Reconstruction. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XL-5/W1, 20133D-ARCH 2013 - 3D Virtual Reconstruction and Visualization of Complex Architectures, 25 – 26 February 2013, Trento, Italy.

- Feng, Ch., Taguchi, Y. and Kamat, V. R., 2014. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. *IEEE International Conference on Robotics and Automation*, DOI: 10.1109/ICRA.2014.6907776, ISBN: 978-1-4799-3685-4, (<https://ieeexplore.ieee.org/abstract/document/6907776>)
- Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. doi:10.1145/358669.358692 (<http://www.cs.columbia.edu/~belhumeur/courses/compPhoto/ransac.pdf>, last visit: 27/12/2016).
- Georgiev, K., Creed, R. T. and Lakaemper, R., 2011. Fast plane extraction in 3D range data based on-line segments. 2011 *IEEE/RSJ International Conference on Intelligent Robots and Systems*.doi:10.1109/iros.2011.6094916
- Hackel T., Wegnerand J. and Schindler K., 2016. Contour detection in unstructured 3D point cloud. *CVPR*, (<https://www.ethz.ch/content/dam/ethz/special-interest/baug/igp/photogrammetry-remote-sensing-dam/documents/pdf/timo-jan-cvpr2016.pdf>, last visit: 3/12/2016).
- Huang, J., and Menq, Chia-Hsiang, 2003. Automatic CAD Model Reconstruction from Multiple Point Clouds for Reverse Engineering. *J. Comput. Inf. Sci. Eng.* 2(3), 160-170, Jan 2003, doi:10.1115/1.1529210.
- Kersten, Th., Büyüksalih, G., Baz, I. and Jacobsen, K., 2009. Documentation of Istanbul Historic Peninsula by Kinematic Terrestrial Laser Scanning. *The Photogrammetric Record*, 24(126): 122-138 (June 2009).
- Krumm J., 2000. Intersection of Two Planes, Redmond WA USA, ([http://research.microsoft.com/en-us/um/people/jckrumm/intersection%20of%20planes/intersection\\_of\\_two\\_planes.htm](http://research.microsoft.com/en-us/um/people/jckrumm/intersection%20of%20planes/intersection_of_two_planes.htm), last visit: 8/02/2019).
- Lu, Chen, and Jianguo, Cai, 2003. Using Vector Projection Method to evaluate maintainability of mechanical system in design review. *Reliability Engineering & System Safety*, Volume 81, Issue 2, August 2003, Pages 147-154, Elsevier Publishing. [https://doi.org/10.1016/S0951-8320\(03\)00075-9](https://doi.org/10.1016/S0951-8320(03)00075-9)
- Lu, Z., Baek, S. and Lee, S., 2008. Robust 3D Line Extraction from Stereo Point Clouds, *IEEE Conference on Robotics, Automation and Mechatronics*, 1-5, 978-1-4244-1676-9/08.
- Rodríguez Miranda, Á., Valle Melón, J. M. and Martínez Montiel J. M., 2008. 3D Line Drawing from Point Clouds using Chromatic Stereo and Shading. Digital Heritage. *Proceedings of the 14th International Conference on Virtual Systems and Multimedia VSMM 2008*. ISBN: 978-963-8046-99-4 pp. 77-84 Point Cloud Library, (<http://pointclouds.org/>, last visit: 18/2/2019).
- Tryfona, M. S. and Georgopoulos, A., 2016. 3D Image Based Geometric Documentation of the Tower of Winds. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLI-B5, 969-975, doi:10.5194/isprs-archives-XLI-B5-969-.
- Vosselmann, G., Gorte, B.G.H., Sithole, G. and Rabbani, T., 2003. Recognizing structure in laser scanner point clouds *International Archives of Photogrammetry & Remote Sensing*, Vol. XXXVI, 8/W2, pp.33-38.
- Yu, Y., Li, J., Guan, H., Wang, Ch. and Yu, J., 2015. Semi-automated Extraction of Street Light Poles from Mobile LiDAR Point-Clouds, *IEEE Transactions on Geoscience and Remote Sensing*, Vol 53, Issue 3.
- Xiao, J., Zhang, J., Adler, B., Zhang, H. and Zhang, J., 2013. Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robotics and Autonomous Systems*, Volume 61, Issue 12, December 2013, pp. 1641-1652, <https://doi.org/10.1016/j.robot.2013.07.001>.