AFTER ABSTRACTION, BEFORE FIGURATION

Exploring the Potential Development of Form Re-topology and Evolution Reapplication with Three-dimensional Point Cloud Model Generation Logic.

CHUN MAN TU[1] and JUNE HAO HOU[2]
[1,2]*National Chiao Tung University*
[1,2]*{chunman|jhou}@arch.nctu.edu.tw*

Abstract. In the era of three-dimensional (3D) informatics, the 3D point cloud modeling algorithm has the potential to further develop. In this study, we attempt to eliminate the limitations of the traditional reverse modeling method and directly turn point cloud data into the material for innovative architectural design by integrating 3D point cloud modeling into the CAD/CAM platform(Rhino/Grasshopper) most widely used by parametric designers. In this way, the randomly ordered point cloud model can be regenerated and reordered according to the designer's requirements. In addition, point cloud data can be spatially segmented and morphologically evolved according to the designer's preferences to construct a 3D model with higher efficiency and more dynamic real-time adjustment compared with the triangular mesh model. Moreover, when a computer vision technique is integrated into the point cloud design process, the point cloud model can be further used to more efficiently achieve rapid visualization, artisticization, and form adjustment. Therefore, point cloud modeling can not only be applied to the spatial structure presentation of building information modeling(BIM) but also can provide further opportunities for creative architectural design.

Keywords. Three-dimensional Point-cloud Model; Computer Vision; Point Set Registration; Topology Optimization; Regeneration.

1. Introduction

The point cloud is a relatively new geometrical class, which is only introduced in the 2000s (Rusinkiewicz and Levoy 2000). Along with the advent of the era of popular photography and the mature development of photogrammetry, the way of capturing physical objects and space into the three-dimensional(3D) data set is limited to high-end devices such as the laser scanners. To date, digital photos taken by ordinary cameras and mobile phones can be used to obtained 3D point cloud data through photogrammetry software with multi-perspective 3D reconstruction principle. At the same time, the textures in images can also be

RE: Anthropocene, Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020, Volume 2, 517-526.

merged into point cloud. As a result, the three-dimensional point cloud data has become a design material that anyone can afford. Nonetheless, Rhino 3D modeling software collaborating with Grasshopper plug-in is now the most widely used parametric design platform in the field of architectural design. This methodology utilizes NURBS(Non-Uniform Rational B-splines) curves as a core technology to generate smooth and flowing curve architecture. However, when using the mesh model that can present convex/concave surfaces and textures' details on this platform, it is not only difficult to edit but also significantly reduce the computing performance. Conversely, the point cloud model not only does not need to record point-to-point sequence relation when generating surface of mesh model, nor does need to calculate the data of huge triangular surface. By storing specific data such as coordinate, time, intensity, scan angle in a single point, highly complex forms in the real world can be realized in Rhino 3D and the result is a faster and more detailed expression. Subsequently, this study endeavors to make point cloud as one of materials for topology and optimization of an early stage of parametric design. We explore how the point cloud data can collaborate directly with python in Grasshopper according to user requirements and design strategies. Different methods of data simplification, data extraction, range selection, and other big data processing are carried out.

This paper incorporates the design logic that architectural designers often used to transform a two-dimensional(2D) plannar models and three-dimensional(3D) stereoscopic model back and forth into point cloud model analysis process. Through the establishment of depth images and conversion of red-green-blue values based on depth data, we can quickly generate 2D images from 3D scenes. Additionally, feature extraction is performed by directly using the highly developed 2D images technique and packages(OpenCV and scikit-image). After that, the 2D images are returned to the 3D coordinates after the morphological transition by using the relationship of indexing and point set registration between pixels and 3D coordinates, and the point cloud model is constructed in terms of stylization, abstraction and morphological evolution. Therefore, the real space or object can be reinterpreted and stimulate more possibilities for development. It also allows design to incorporate both aesthetic and practical requirements simultaneously, making it be quickly customized according to the physical models and objects in the modern industrial era. It has also developed into a potential design material for diverse applications.

2. Discussion

Color is an integral element of our world, not just in the natural environment but also in the man-made architectural environment. Color always played a role in the human evolutionary process(Frank H. Mahnke). Point cloud data are obtained from immense databases of 3D coordinates and specific image attributes (e.g., color, intensity, and brightness) produced through 3D scanning and photogrammetry. Thus, designs via point cloud can more comprehensively analyze spatial contexts and can incorporate diverse space-time factors such as color and lighting instead of focusing only on the architectural modeling in XYZ space. 3D point cloud applications have increased rapidly in recent years.

From the perspective of the cross-platform Open Source Computer Vision Library (OpenCV), the trend of relevant modules being utilized for the processing of 3D point cloud data has grown gradually, especially in the translation of 2D imagery to 3D vision. Notably, the use of Rhino/Grasshopper makes the form of editing geometries computationally simple and easy to automate through mesh vertices, especially when modeling an irregular shape or designing a project consist of many complex elements in the site. Moreover, the program performs well even when the surface of the mesh model is removed and only color-coded vertices are retained. Therefore, we have explored the additional possibilities and innovative potential of integrating related open-source python libraries and sending required data to the CAD/CAM platform (Rhino/Grasshopper).

2.1. LIMITATIONS AND IDIOSYNCRASIES OF MESHLAB

MeshLab, an open-source 3D mesh-processing software, is used to develop multiple filters and regenerative methods. However, for designers, it is difficult to master this program without a strong computer-science background. When considering the colors of a 3D model, there are two options for visualizing and controlling MeshLab's encoding. One method requires the use of a color vertex. The other requires the use of texture mapping. The color vertex method is much easier to use. However, to retain color detail, a model comprising very small triangles must be saved. This comes with huge computational costs. Thus, most users prefer texture mapping. However, color data are not encoded in a geometrical context via this method and are instead included in an external file. Thus, users must clearly understand parameterization. Otherwise, when deleting invisible layers or re-meshing, the color information can be lost.

2.2. POTENTIAL OF THE DURABLE ARCHITECTURAL KNOWLEDGE (DURAARK) PROJECT

DURAARK was funded through the European Commision's FP7 Programme from 02/2013 to 01/2016. CloudCompare, a 3D point cloud processing software based on OpenGL, was used for development. It relies on an octree structure that is highly optimized for huge point clouds using color information. It also performs many simple analyses. It does well visualizing large point cloud data, and it provides a strong data analysis function. In terms of design, it lacks adjustment flexibility with editing points in defined regions having irregular boundaries. Furthermore, the point picking function of CloudCompare is quite slow. However, CloudCompare has performed very well in popular model analyses, but it has been weak for detailed designs or form evolution platforms. Therefore, this project develops an excellent plug-in volvox for Grasshopper to solve multiple problems of point cloud integration for architectural modeling programs, including reading and writing various file formats of point cloud data and merging or subsampling point cloud models. However, the entire project focuses more on form regeneration and evolution of the overall model. Thus, meshing the point cloud via voxelization is the only way to achieve form re-typology. Therefore, this project examines the potential of using point cloud data as design material and uses color and the change in normal vectors as reference data to discover new and creative approaches for re-typology. It is expected that the form evolutions and the detailed editing of specific areas of the point cloud will be flexible for both models for the highly popular Rhino/Grasshopper visual programming environments.

2.3. ENHANCING THE DIVERSITY AND CREATIVITY OF THE POINT CLOUD MODEL DESIGN BY INTEGRATING THE OPEN SOURCE LIBRARY OF RHINO / GRASSHOPPER

Point clouds are sources of realistic scenes and virtual objects that can be used in creative ways. Thus, we have attempted to incorporate 2D planes, which designers often use during the design stage for point cloud analysis and design. During this process, the point cloud model generated by professional photogrammetry software is used to create depth images through plane projection. The depth information is then converted to red-blue-green values to quickly support image ranges. Image processing techniques (e.g., OpenCV and scikit-images) are well developed used to complete edge extraction, key-point detection, material replacement, object merging, and adjustment. Point set registration is used to convert an image pixel to back to 3D. Thus, we can retrieve results with high efficiency and accuracy. This process also provides designers point cloud design concepts so that they can build a smoother communication platform between designers and customers.

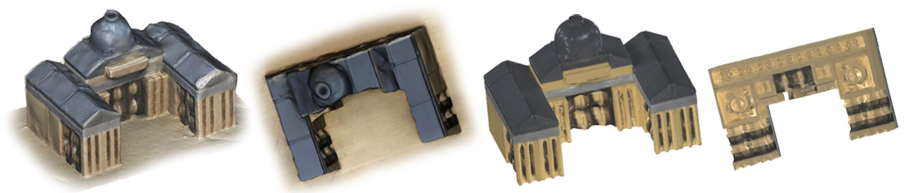
3. Methods

Point cloud data enable media to accurately present information regarding material and spatial relationships in the real world and substances on a virtual platform. Since there is no sequential relation between points and they are not related to each other, we attempted to eliminate the complex processes and massive calculation memory required for editing the high-poly mesh model in Rhino/Grasshopper. Moreover, OpenCV(open-source computer vision) is one of the most popular, and powerful libraries in image processing. Using python-remote-control(a plug-in for Grasshopper) to transmit the required data from Python IDE to Grasshopper, the issue of cross-platform data transfer can be easily resolved. Therefore, we aimed to develop a design process for the 3D point cloud model's morphology analysis, generation, and optimization based on the open-source library(OpenCV), and CAD/CAM platform(Rhino/Grasshopper). In doing so, the difficulties of disordered point cloud re-typology, such as selecting/deleting a specific range of points, feature detection, texture adjustment/replacement, and form evolution, can be efficiently resolved, and the process of point cloud editing will become more designer-friendly. So, that point cloud model is used directly for re-topology, morphogenesis, which meets the requirements of customized and refined design. The processes are detailed in the following sections.

3.1. DATA CAPTURING: EXPLORING EFFICIENT METHODS TO OBTAIN A 3D POINT CLOUD MODEL FROM DIFFERENT FORM AND SCALE OBJECTS

Photogrammetry and 3D scanning are the two primary methods of reverse modeling from a real object. This project attempts to utilize red-green-blue values and normal vectors from the point cloud to investigate the potential of the evolution of the form of the 3D model. Therefore, the two methods mentioned above are employed for the reconstruction of the digital 3D model, and the obtained results are analyzed to ascertain efficient means of attaining superior modeling outcomes in discrete situations to allow compliance to the requirements of subsequent design processes. The results(see figure 1) of the study are as follows:

- Photogrammetry reconstructs models by comparing pixel colors and defining the anchor points of 2D images. The lack of restrictions in terms of the object scale is an advantage of this method, and its disadvantage is that a completely closed mesh model cannot be obtained, because the object is not allowed to move during the photo-taking process. Errors are also common with smooth objects that do not evince obvious edge features.
- The use of the non-contact optical laser scanner (Artec Space Spider) is based on the structured-light method of 3D reconstruction, and Artec Studio utilizes features in overlapping areas to automatically align captured frames. Therefore, the position and angle of the object can be changed arbitrarily before each scan to obtain completely closed mesh models with details. However, a focus problem causes the 3D laser scanner to be unsuitable for the reconstruction of objects that are too large or too small.



The result of 3D photogrammetric reconstruction | The result of 3D laser scanning reconstruction

Figure 1. Comparing photogrammetry and 3D laser scanning methods.

3.2. DATA REGENERATION: USING VOLVOX (A PLUG-IN FOR GRASSHOPPER) AND PYTHON-PCL TO IMPORT AND RESAMPLE POINT CLOUD DATA

The resample module, which is based on the moving least squares function(MLS) in the point cloud library (PCL), was used to estimate the normal vector of points, and improve issues such as broken, and rough surfaces caused by shadows or light reflection. Then, we proposed the use of RANSAC (Random Sample Consensus) through the sample consensus module in the PCL to remove outliers and noise near ground. Thus, the data obtained using photogrammetry can be regenerated without noise(see Figure 2).

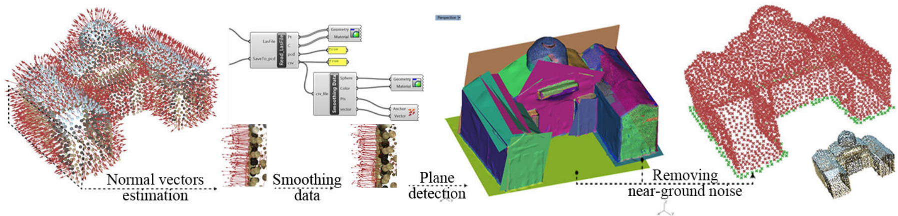


Figure 2. The process of point cloud model regeneration in Rhino/Grasshopper.

3.3. DATA INDEXING: TRANSFORMATION FROM 3D WORLD COORDINATES TO TWO-DIMENSIONAL (2D) PIXEL COORDINATES FOR INTEGRATING COMPUTER VISION TECHNIQUES INTO POINT CLOUD PROCESSING

First, we indexed the point cloud and selected the number and location of the necessary projection planes according to form and features. Next, we projected points onto multiple selected planes, and the shortest distance between each point and its projection point was considered the depth data. Therefore, when two or more points were projected in the same position, only the point with the smallest value in the depth data was selected to be projected onto the plane. Moreover, we used the module in OpenCV to establish depth images based on the depth data and the obtained projection result. Finally, we converted depth images to range images(see Figure 3) according to the values in the alpha channel to ensure that the point selection and feature detection in the next step can occur without interference from different materials or surfaces.

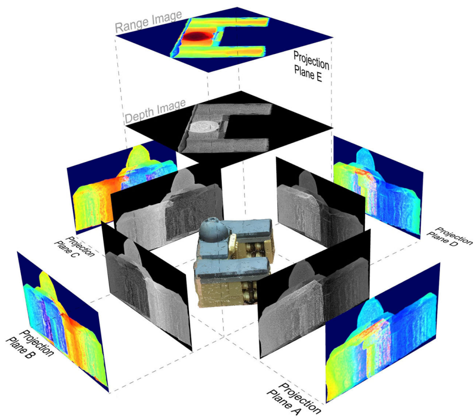


Figure 3. Spatial relation diagram and projection results.

3.4. DATA ABSTRACTING: USING POINT SET REGISTRATION AND A COMPUTER VISION TECHNIQUE TO IMPROVE THE ADJUSTMENT FLEXIBILITY OF THE 3D POINT CLOUD MODEL

We indexed each pixel in the projection image and stored red-green-blue and depth data sequentially in accordance with the image coordinate system. By this approach, we utilized the index relationship between the 3D point cloud and each of the pixels to complete the point set registration process and minimize the data. Furthermore, by combining computer vision and image synthesis technology, we were able to efficiently adjust the projection planes by regenerating the feature points/edges and replacing the material and then returning the results to the 3D point cloud model using the point set registration method. For example, we can use the "Sobel edge detection" function in scikit-image library o extract edges and pixel coordinates with an alpha value higher than 200 and use the index relationship that we have established previously to select points near the edges in the 3D point cloud model. Finally, after completion of the registration, we used the "voxelgrid filter" function in python-pcl to voxelize the edges' points to accurately locate the positions of the edges' lines in the model(see Figure 4).

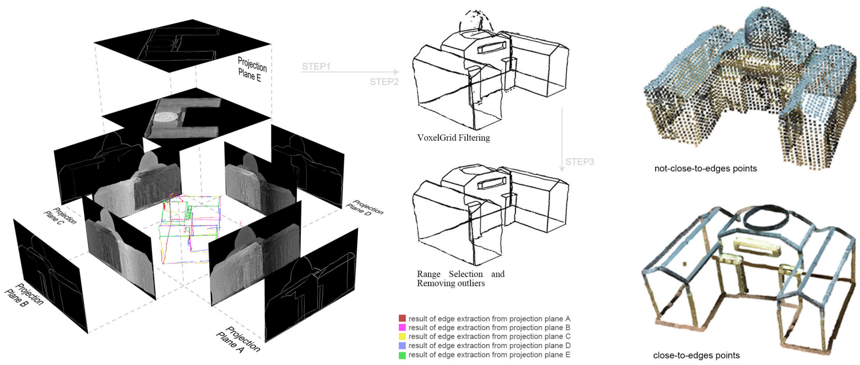


Figure 4. The result after superimposing each projection plane for Canny edge detection and registration.

4. Experiment

The main purpose of this study is to explore how point cloud data with high spatial dimensions can be used as a creative design material and be flexibly applied to diversified design fields. Therefore, we tested the above-mentioned approaches in diversified design fields. For example, we constructed a point cloud model with artistic visual effects, evoluted the form from a cube to furniture, and optimized the detail after form re-topology. The test results are as follows:

4.1. ARTISTICIZATION OF POINT CLOUD MODEL

We re-sampled the data from the projection results to ensure that each of the points in the 3D model was only projected on the specific projection plane with the shortest perpendicular distance from the projection plane. We then imported the results to Photoshop, created replacement materials, and filtered special effects processing. Finally, the results were returned to the 3D model using the point set registration method. Furthermore, material replacement and special visual effect presentation of the point cloud model (see Figure 5) were carried out. By these processes, a design concept can be quickly simulated and visualized in Rhino/Grasshopper, so that rational analysis and logic can be combined with perceptual design thinking and more innovative design results can be presented.

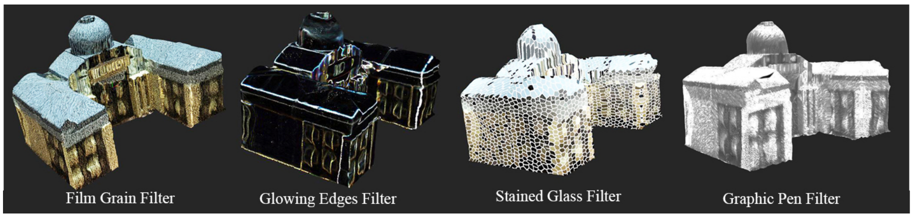


Figure 5. Artisticization of point cloud model.

4.2. APPLICATION OF POINT CLOUDS TO THE FORM EVOLUTION

In this stage, we used a simple cube to explore the potential of the 3D point cloud model typology using the hybrid modeling approach that we proposed in order to prove that making a combination between the 3D models in Rhino/Grasshopper can be creative and at the same time efficient as the 2D image collage of the different elements in Photoshop. The results are as follows:

4.2.1. Selecting an irregular boundary area of points for replacing red-green-blue values and translation of points

First, we projected all the points in the 3D model onto the nearest projection planes and attached the patterns with highly complex shape onto the projection images(see Figure 6: step 1). We then used the point set registration approach to select the points which were covered by the patterns in the projection images by the index relation that has been established previously. Finally, each selected point's red-green-blue value was substituted(see Figure 6: step 2), and the position was moved to a certain distance along the normal vector according to the alpha value of the corresponding pixel on the pattern(see Figure 6: step 3 to step 5), so that we can prove that the 3D point cloud model can promptly and comprehensively select a specific area and regenerate using the selected index of pixels in the projection images.

4.2.2. Form evolution through depth images

Each point in the point cloud model is independent and is not affected by the neighbors, In this context, the point cloud model requires the attachment of the depth image of the object to be combined with the original model in the most suitable projection image, and it uses the module in OpenCV to extract the alpha channel value in the area. As a result, the value will be directly projected onto the corresponding point in the 3D point cloud model. The depth images can also be used to extend or copy the existing points to produce smooth curves/surfaces and other effects(see Figure 6: step 7 to step 9). Thus, the design can meet the functional requirements and demonstrate modeling aesthetics accordingly.

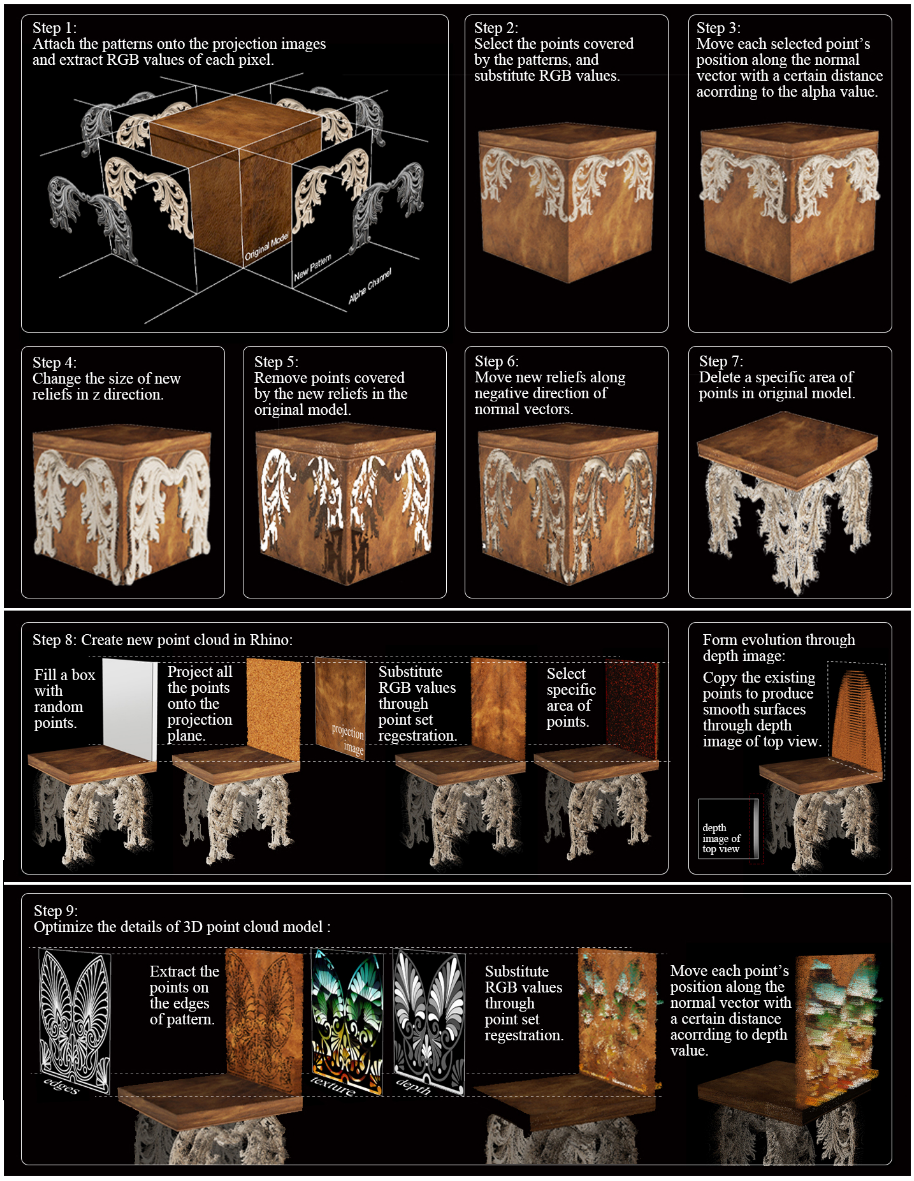


Figure 6. The process of form evolution through 3D point cloud registration algorithm.

5. Conclusion

We integrated multiple open-source python libraries into our pre-design process and sent the required data to CAD/CAM platform(Rhino/Grasshopper) with the aim of representing point cloud data analysis techniques and algorithms that have already been well developed from the designer's perspective. The application of rapid modeling will not only be pursued for the practical expression of realistic senses in virtual scenarios, but the said models will also be equipped with human-specific perceptual thinking, and creativity capabilities. By re-analyzing those techniques and mathematical rules, creative inspiration beyond rational logic might be induced and thus allow architectures to not only impart and inherit history but also create future opportunities to march toward a better living experience.

6. Further Steps

Rhino/Grasshopper, despite being an excellent platform for fast testing and rapid visualization, is inferior to game engines in overall computational efficiency in terms of point cloud processing of extremely big data. In contrast, the design of the overall analysis demonstrated in this study allows the integration into Blender with C-Python as a platform. This technique should not only be applied in preliminary architectural design and analysis, but it should also be used in design simulation and art with even bigger data derived from drone photogrammetry. Moreover, regarding the planar projection that we used in this research, there are relevant problems of anamorphosis or distortion when using the original model; the surfaces of which are not flat and edges not bent. Therefore, the UV mapping coordinate setting rules in OpenGL can also be adopted in the future to flatten the point cloud model, which can further improve the accuracy of optimization, and re-typology models using computer vision technologies.

Acknowledgment

This research was supported by the Ministry of Science and Technology under Grant Number 109-2634-F-009-015, and was managed by the Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan.

References

"opencv-python" : 2000. Available from Open Source Repository<https://github.com/skvark/opencv-python>.

"python-PCL(Point Cloud Library)" : 2017. Available from Open Source Repository<https://github.com/strawlab/python-pcl>.

Achlioptas, P., Diamanti, O., Mitliagkas, I. and Guibas, L.: 2017, Learning representations and generative models for 3d point clouds., *arXiv preprint arXiv:1707.02392*.

Landrieu, L. and Simonovsky, M.: 2018, Large-scale point cloud semantic segmentation with superpoint graphs, *IEEE Conference*, 4558-4567.

Mehra, R., Tripathi, P., Sheffer, A. and Mitra, N.J.: 2010, Technical Section: Visibility of noisy point cloud data, *Computers and Graphics*, **Volume 34 Issue 3**, 219-230.

Zwierzycki, M., Evers, H.L. and Tamke, M.: 2016, Parametric Architectural Design with Point-clouds: Volvox, *Proceedings of eCAADe 2016*, Finland, 673-682.