



# A SINGULAR WORKFLOW FOR 3D SURFACE RECONSTRUCTION OF HEAVILY NOISY POINT CLOUDS

Nebojša Nešić\*,  
Miloš Stojmenović

Singidunum University,  
Belgrade, Serbia

## Abstract:

Surface reconstruction from low quality point clouds represents a common problem in most standard algorithms created for this purpose. Point clouds acquired using specialized devices, such as 3D scanners, or as outputs from structure from motion algorithms are usually flawed in that they contain a significant amount of noise and outliers, making the surface reconstruction process difficult, resulting in low quality surface estimation. The quality of the reconstructed mesh is directly proportional to the quality of the point cloud itself. This paper proposes a workflow for creating 3D surfaces from unstructured point clouds. The workflow takes an unstructured point cloud as input and, through four phases, automatically cleans up the point cloud data and creates a watertight surface reconstruction of the point cloud, all in a single, end-to-end workflow.

## Keywords:

Surface Reconstruction, Point Cloud, Outlier Removal.

## INTRODUCTION

Raw point cloud data acquired using specialized devices such as LiDAR sensors or as structural outputs from motion algorithms that use RGB images and depth data along with feature matching, usually contain a large amount of noise and outliers. Point clouds in such a state cannot be easily reconstructed into a 3D mesh, but rather must first undergo a process of data cleaning. As such, the process of generating the 3D mesh from a raw point cloud can be separated into two distinct phases, the first phase being clean-up and the second phase being the reconstruction itself.

The amount of data clean-up must be carefully constructed in order to preserve as many features and details as possible, while removing as much noise as possible. The subject of de-noising has been extensively studied and adapted for different surface types, object shapes and other parameters [1]. There is no single method that prevails above all others. Every method has applications in certain use cases. Unfortunately identifying the best method for a particular point cloud is usually experimental in nature.

## Correspondence:

Nebojša Nešić

## e-mail:

nnesic@singidunum.ac.rs



Similar to noise removal, the subject of surface reconstruction from 3D point clouds is not a new area of research. As such, there are many algorithms to choose from when performing surface reconstruction, each with their own pros and cons [2].

The choice of algorithm is invaluable in getting satisfactory results, due to the fact that each algorithm handles specific drawbacks of non-perfect point clouds in different ways.

## 2. RELATED WORK

As mentioned in the introduction, both phases of the reconstruction process have been extensively researched. We will briefly review the methods that influenced and motivated our work below.

The removal of outliers has many approaches. Sun et al. [3] proposed a method for removing outliers that utilized L0 minimization, thus creating sparser outputs and retaining sharp features and surfaces, thus eliminating an recurring issue of oversmoothing in outlier removal and surface reconstruction. However, sparse outputs will not retain as many surface details as dense outputs would. A different approach is using neighbouring points in a point cloud to detect clusters. These approaches are usually slow in computation, where the speed is dependant on the complexity of the data. Sankaranarayanan et al. [4] described an algorithm that improved the computational speed in using neighbourhoods of points in outlier removal. Ning et al. [5] removed noise from point clouds by analyzing categories of outliers identified in the cloud itself. Due to using the local density of points for detecting outliers, this method is not suitable for point clouds where dense clusters of outliers exist. Yuan et al. [6] developed a novel method called spatial neighbourhood connected region labelling, which is used for data clustering in the point clouds. The method was tested on both synthetic and real world point cloud data. The points are classified into clusters which are then used for identifying cluster outliers as well as sparse outliers. Li et al. [7] extracted the nearest neighbouring points before the search began, thus removing repetition with the Euclidean distance calculation that is required, and saving time and resources. Reconstruction of point clouds usually relied on standardized, tried and tested algorithms, such as [8] and [9]. These algorithms are heavily dependent on the input and as such the quality of their output is directly affected by the outlier removal. With the development of machine learning, approaches that utilize more complex

workflows for surface reconstruction have begun to appear. Hanocka et al. [10] uses a deep neural network to deform an initial mesh to encapsulate the input point cloud, thus ensuring the shape of the object does not lose the actual point cloud shape. The authors demonstrated the robustness of the algorithm on point clouds of non-trivial shapes that vary in density and quality, showing that the algorithm outperformed multiple standard algorithms such as [8]. Ladicky et al. [11] propose a surface reconstruction method that utilizes regression forests, using predictions that depend on the local context. The training of the network was done on synthetic data, which represents the biggest problem of the pipeline itself, due to the difficulty of generating quality noisy point cloud data that mimics realistic scenarios. That, in itself, represents the biggest issue that the neural network approaches have - the quality of data acquisition. Dill et al. [12] presents a generative model that progressively deforms a uniform sphere mesh until it approximates the input point cloud. The limitations of this approach are its limitations to objects with a similar topology as well as failure to preserve the finer detail.

## 3. PROPOSED METHOD

Our solution combines the process of outlier removal with the process of surface reconstruction to create watertight 3D meshes. The point cloud data used to showcase the workflow is deliberately of a lower quality, with a large amount of internal and external noise [Figure 1]. The process of creating the 3D surface reconstruction of this point cloud consists of four phases.

In the first phase, initial outliers are identified and removed from the input point cloud. Only the most extreme outliers are removed in the first phase. Initially the centre of mass of the point cloud is calculated, after which the points that are located furthest away from the centre are identified as the initial candidates for removal. A k-nearest neighbour is implemented to detect a neighbouring area around the point. Once the area is selected, the number of points inside the area are counted. If the number of points is lower than the threshold value, the points are identified as outliers and removed from the point cloud [Figure 1]. The threshold is variable and depends on the distance of the point from the centre of mass of the point cloud.

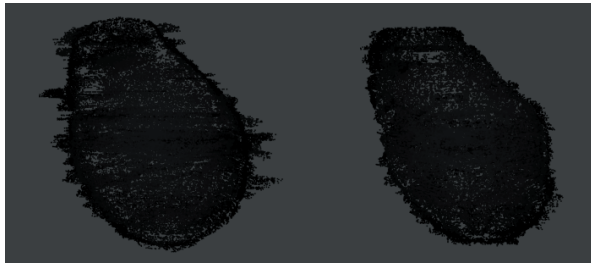


Figure 1 – Noisy input point cloud (left);  
after initial noise removal (right)

After the removal of the most extreme noise in the point cloud, a rough approximation that envelopes the entire point cloud is generated. The approximation is generated using the Convex Hull approach. Once this initial mesh is calculated, both the mesh and the point cloud are loaded into the same environment [Figure 2]. Due to the fact that the Convex Hull does not take cavities into account when generating the mesh, the initial mesh needs to be deformed in order to follow the shape of the original point cloud. In order to do so, the initial mesh is projected directly onto the surface of the point cloud. However, a Convex Hull mesh is not prone to quality deformation. To alleviate this, a re-meshing of the Convex Hull is performed. The purpose of this step is to create an evenly distributed face topology that will allow the mesh to deform uniformly across the entire object [Figure 3]. Once this step is complete, the projection of the mesh may be done. The main attribute of the mesh that dictates how much the mesh can deform is the amount of polygons that it consists of. In this step, the number of polygons is not increased as the purpose is not to go into minute details, but to create a more precise shape that follows the contours of the point cloud [Figure 4].

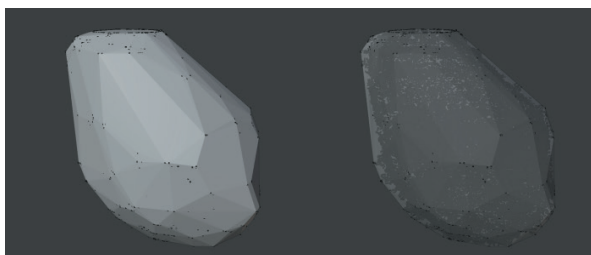


Figure 2 – Convex Hull (left) ;  
Convex Hull with point cloud (right)

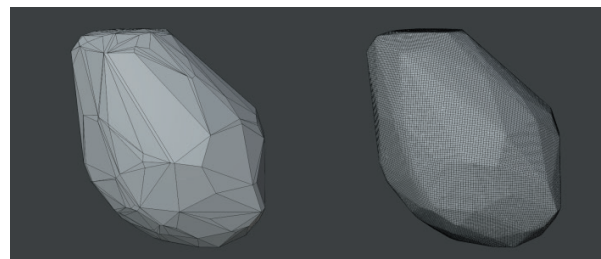


Figure 3 – Convex Hull initial topology (left);  
Convex Hull after re-meshing (right)

In the third phase, the rough estimated mesh is used to identify the points near the contour of the object that still represent noise and reduce the number of surface details that can be reconstructed. Both the rough estimation and the point cloud are again loaded into the same environment, where now there should be no points that are outside the mesh. The mesh is then scaled by a miniscule amount iteratively, and the number of points that are exposed with each scaling iteration are counted. A threshold is defined as a percentage of the density of the point cloud itself. For each iteration, if the number of exposed points is under the defined threshold, the points are identified as noise and removed from the point cloud. Once the number of exposed points reaches the threshold, the process is halted, and a new point cloud is defined [Figure 5].

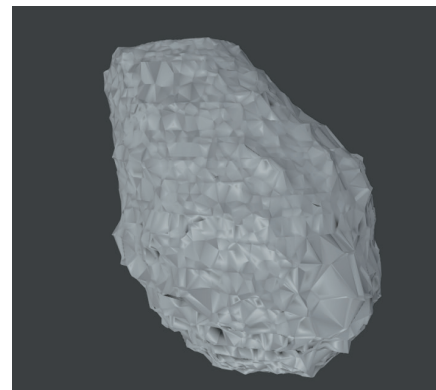


Figure 4 – Rough shape estimation of the point cloud



Figure 5 – Point cloud after the second round of noise removal

Once the final point cloud is created, the process from the second phase is repeated albeit with some modifications. The Convex Hull of the point cloud is calculated again, and the meshing process is repeated again. This time, however, the resulting mesh is subdivided into a more dense mesh i.e. a mesh that contains a higher number of polygons and as such can be deformed to reconstruct more of the finer details. Again, the resulting subdivided mesh is projected onto the point cloud [Figure 6]. The more cavities the point cloud contains, the higher the subdivision level should be. After this step, the mesh is finalized and ready for use. A smoothing of the surface can be implemented if necessary, but was not included in our solution.

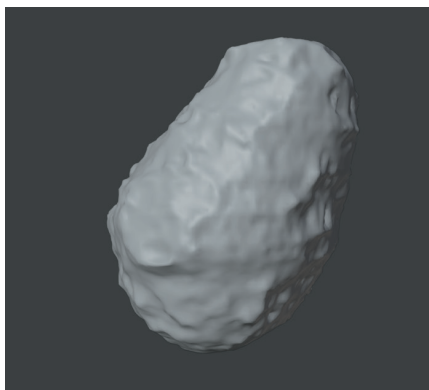


Figure 6 – Final version of the surface reconstruction

## 4. RESULTS

The resulting mesh is compared with standard approaches for reconstruction. To test out the quality of the reconstruction, the input point cloud in all of the reconstruction algorithms is the final version of the point cloud from the workflow, the use of which ensures uniform conditions that will only focus on the reconstruction process.

In [Figure 7] the results of the reconstruction are shown. It is clearly visible that the standard algorithms did not deal with the varying density of the point cloud well, creating large gaps in the surface, and small cavities where there should be none. Outlier and noise removal will only go so far in improving the quality of the point cloud. In some cases, as it is the case here, the point cloud will have parts of the surface missing. Combined with the internal noise that is present, this could create an illusion of gaps and cavities that should not be present. The workflow presented in this paper deals with all of the mentioned issues, and creates an accurate representation of the point cloud in question.

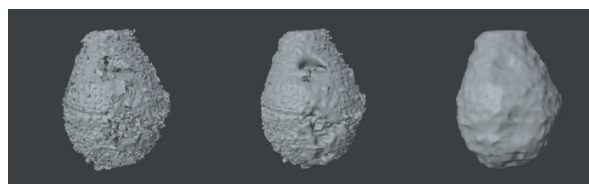


Figure 7 – Marching cubes (left);  
Poisson surface reconstruction (middle);  
our workflow (right)

## REFERENCES

- [1] X.-F. Han, J. Jin, M.-J. Wang, W. Jiang, L. Gao i L. Xiao, „A review of algorithms for filtering the 3D point cloud,“ *Signal Processing: Image Communication*, t. 57, 2017.
- [2] S. P. Lim i H. Haron, „Surface reconstruction techniques: A review,“ *Artificial Intelligence Review*, t. 42, 2012.
- [3] Y. Sun, S. Schaefer i W. Wang, „Denoising point sets via L 0 minimization,“ *Computer Aided Geometric Design*, tom. %1 od %235-36, 2015.
- [4] J. Sankaranarayanan, H. Samet i A. Varshney, *A Fast k-Neighborhood Algorithm for Large Point-Clouds*, 2006.
- [5] X. Ning, F. Li, G. Tian i Y. Wang, „An efficient outlier removal method for scattered point cloud data,“ *PLOS ONE*, t. 13, 2018.



- [6] X. Yuan, C. Huawei i B. Liu, „Point cloud clustering and outlier detection based on spatial neighbor connected region labeling,“ *Measurement and Control*, 2020.
- [7] Z. Li, G. Ding, R. Li i S. Qin, „A new extracting algorithm of k nearest neighbors searching for point clouds,“ *Pattern Recognit. Lett.*, t. 49, pp. 162-170, 2014.
- [8] M. Kazhdan, M. Bolitho i H. Hoppe, „Screened Poisson Surface Reconstruction,“ *ACM Transactions on Graphics*, t. 32, pp. 61-70, 2006.
- [9] W. Lorensen i H. Cline, „Marching Cubes: A High Resolution 3D Surface Construction Algorithm,“ *ACM SIGGRAPH Computer Graphics*, t. 21, p. 163, 1987.
- [10] R. Hanocka, G. Metzer, R. Giryes i D. Cohen-Or, „Point2Mesh: a self-prior for deformable meshes,“ *ACM Transactions on Graphics*, t. 39, 2020.
- [11] L. Ladicky, O. Saurer, S. Jeong, F. Maninchedda i M. Pollefeys, *From Point Clouds to Mesh Using Regression*, 2017.
- [12] A. Dill, C. Li, S. Ge i E. Kang, „Getting Topology and Point Cloud Generation to Mesh,“ *ArXiv*, t. abs/1912.03787, 2019.