

STL Generation from Point Cloud Data with User-Controlled Triangulation for Additive Manufacturing



Sourabh Chasta, Narendra Kumar, Vishal Francis, and Prashant K. Jain

Abstract In the last two decades, reverse engineering (RE) and additive manufacturing (AM) have been emerged together as a tool to fabricate intricate parts in minimum time. RE captures the point cloud data and converts it to standard triangulation language (STL) which is compatible to AM systems. The existing approaches to generate STL either use costly software or use decade-old algorithms, which are ambiguous and difficult to understand. The current research proposes a mesh construction algorithm (MCA), which endeavors to provide a detailed, simplified and modified algorithm to generate STL from unorganized point cloud data using Delaunay triangulation and its properties. An additional feature of proposed MCA is that user can control shape of mesh, and chooses best STL according to the requirement. Moreover, this article describes a descriptive algorithm to create STL from cloud data in easy way using MATLAB and its function. The generated STL files are compatible with other software and parts can be 3D printed directly using these files.

Keywords Additive manufacturing · Reverse engineering · CAD · Mesh construction algorithm

S. Chasta · P. K. Jain

Mechanical Engineering Discipline, PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, India

N. Kumar (✉)

Industrial and Production Engineering Department, Dr. B R Ambedkar National Institute of Technology, Jalandhar, India

e-mail: kumarn@nitj.ac.in

V. Francis

School of Mechanical Engineering, Lovely Professional University, Jalandhar, India

1 Introduction

Additive manufacturing (AM) is an emerging domain of engineering, which has gained popularity due to ability of fabricating complex objects directly from CAD model. AM provides freedom to affordable manufacturing of a complex shape in less time. It reduces the lead time of product development cycle by providing a prototype prior to start actual manufacturing. Before coming into hands, the produced prototype from AM has to pass through various steps in a chained system. Generation of CAD model is the primary step and can be performed through various software. Sometimes, the available software for modeling shows incompetency in creating CAD models of the intricate shapes like architecture buildings, unsymmetrical artifacts, human bones, etc. The modeling of these shapes in dedicated software may be time consuming and uneconomical. In this situation, reverse engineering (RE) technique is utilized to generate the CAD models of unsymmetrical and complex geometries. It collects physical data information of an object via scanning operations using contact or non-contact type machines. The coordinate measuring machine is commonly used in contact type scanning machine, whereas laser scanners come under the non-contact type scanning machines. These measuring machines provide coordinates of an object, which is commonly referred as point cloud data. The point cloud data is further processed and converted in a supportive mesh file for additive manufacturing.

The mesh file format should support both modeling and printing software or it should be written in a universally accepted format. Standard triangulation language (STL) file format is used as the de facto standard for mesh files. It stores only surface information of the 3D object described as an unstructured triangulated surface. The conversion of point cloud into STL is a complex and difficult task, but there are some commercial software like Geomagic Control, point cloud processing of 3DReshaper, etc., available in the market to accomplish this purpose. Moreover, there are some software freely available but they are less efficient in generating defect free STL. However, sometimes there are compatibility issues occurred during the STL exchange with other software. Thus, more research needs to be carried out for developing an economical and compatible platform to generate the error free STL directly from the point cloud.

Thus, the present paper aims to develop an algorithm to generate the STL file via proposed mesh construction algorithm (MCA) with user-controlled tessellation. The MCA can be used reverse engineering, virtual reality and computer vision, moreover, it can be used to integrate with additive manufacturing. The paper also provides a methodology to imply MCA in MATLAB software, which is mostly used by researchers for its advanced technical computing and user-friendly interface. The generated STL files are fabricated using 3D printer in order to validate the present work.

2 Literature Review

Construction of surface for 3D point cloud data was considered as an impossible task in past decades. The fortune, sweep line and Delaunay algorithms were confined to solve only 2D mesh problems. In the recent years, some studies proved the possibilities of generation of mesh for 3D data. Choi et al. [1] presented an optimal triangulation algorithm for 3D discrete points with smoothness surface criterion using Delaunay triangulation (DT). Bernhard et al. [2] proposed a solution to create 3D surface using DT for planar surface and create tetrahedron between two consecutive planar surfaces. They reconstructed the surface of bone using scanned images. Varady et al. [3] identified the RE in industrial applications and discussed about the problems in data acquisition technique and surface construction from wrong data points. Piegler et al. [4] presented a method to reconstruct the model from point data using B-spline curves by approximate the data and controlling the tolerances of knot vectors. Recently, Woo et al. [5] presented a novel method to segment the scanned data by using octree-based 3D grid technique and even works on unorganized data points. After successful creation of 3D surface from scanned data, researchers utilized the potential of RE in additive manufacturing by converting scanned data directly into STL. Chen et al. [6] presented an approach to generate STL file from coordinate measuring machine (CMM) with user control point reduction. Their study was mainly focused on data reduction of point cloud and was limited to create STL for organized data points. Lee et al. [7] developed an algorithm to generate STL by dividing scanned data in smooth, rough and plane regions. Nevertheless, the study showed the generation of STL file using Christiansen algorithm [8] and focused more on refining the prepared STL using data reduction. Recently, more advanced work has been attempted by researchers in which one more step has been reduced in RE technique by directly slicing the point cloud to 3D print the part. Xu et al. [9] presented an improved virtual edge approach to slice point cloud data and form contours. Their focus was on improving the quality of contours by filling the gaps in point cloud by providing additional points. Piegler et al. [10] presented a work on the direct slicing of point cloud. In their approach, slicing and printing were done simultaneously for one layer then another set of data points were considered to prepare next slice. Yuan et al. [11] presented the direct slicing approach to print part contours directly from point cloud without preparing any CAD model or STL. The printing of complex geometries from point cloud using FDM printers was performed. Their approach of part printing was not versatile in nature as it failed to create toolpath for area inside the contours. PLSP-based approach has also been used to develop contour information from point cloud for additive manufacturing. The method proposed is robust as it does not involve the nonlinear optimization [12].

Aforementioned literature shows that various approaches have been presented to convert scanned data into 3D surface or STL. The presented algorithms that generate STL are complex in nature and outdated to process large number of data points. Therefore, present paper aims to develop a modified algorithm of STL generation from any kind of 3D scanned data, either organized or unorganized.

3 Terminologies

- Point Cloud: Group of points situated anywhere in three-dimensional space referred as point cloud. The point clouds refer to the surface points obtained from 3D scanners and coordinate measuring machines.
- Circumsphere and circumcircle: A sphere that touches all vertices of polyhedron and is bound in its shape known as circumsphere. Similarly, circumcircle is used for planar case. Commonly, both terms are used in DT in context of tetrahedron and triangles meshing. The radius of circumspheres or circumcircles is known as circumradius.
- Alpha Shape function: It is an in-built function in MATLAB used to generate triangulated mesh over point cloud data. It uses DT to tessellate the point cloud. It uses alpha radius to construct mesh, which is equivalent to circumradius.
- Boundary Facet function: It is also an in-built function used to generate triangular connectivity list for the given 3D point cloud.

4 Methodology

In order to generate STL, a certain set of points is required. Therefore, in the current study, the point cloud data acquired from various resources are stored in the form of matrix. In this matrix, rows represent the number of points, whereas corresponding coordinates are represented in the columns. Based on this matrix, the mesh construction algorithm (MCA) distinguishes the nature of point cloud and then proceeds further to form a mesh. A well-known Delaunay triangulation (DT) property has been used to construct a mesh over data. Subsequently, STL file has been prepared with the help of the constructed mesh. The aforementioned method of preparing STL is implemented in MATLAB software. The prepared STL is then analyzed through exporting in Solidworks software. After ensuring the quality of STL, final validation has been done using 3D printing software and printer. Physical prototypes are printed using prepared STL to demonstrate the robustness of developed algorithm. The methodology adopted in the current research work is shown in Fig. 1.

Delaunay triangulation is used to tessellate the discrete point in a three-dimensional plane. DT creates tetrahedron for 3D discrete points and triangles for planar points. All tetrahedrons and triangles must follow the Delaunay's property that no other points should fall inside the circumscribed sphere or circle. For 3D, DT creates tetrahedron using 4 discrete point circumscribed in sphere. General equation of sphere in 3D plane is $(x - a)^2 + (y - b)^2 + (z - c)^2 = r^2$, where (a, b, c) is center of sphere and (x, y, z) is any point on the surface of sphere. The unknown parameters used in this equation can be determined using the following steps:

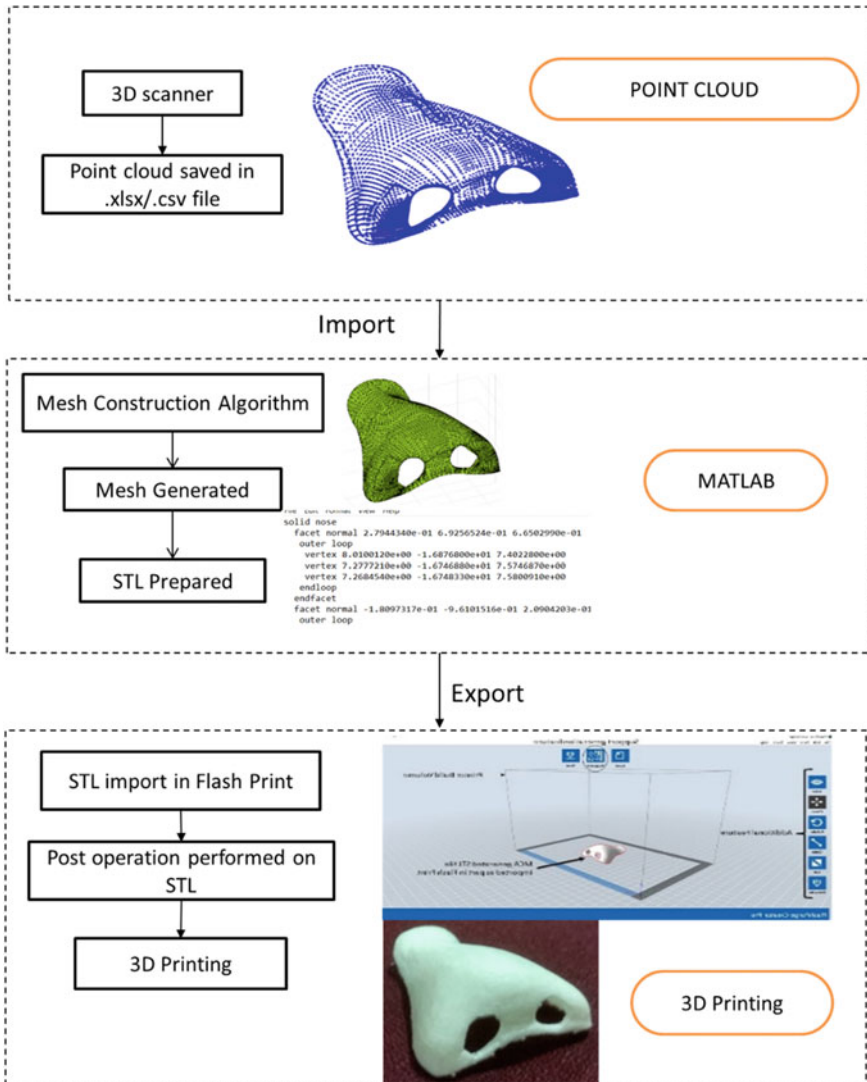


Fig. 1 Methodology

- Select any four points $\{P1, P2, P3, P4\}$ from cloud data and create sphere using above equation. When sphere radius and center are known, check the property of Delaunay on it.
- Check the Delaunay property on circumsphere by satisfying following condition then only tetrahedron will be considered valid.

$$\forall \{x, y, z\} - \{P1, P2, P3, P4\} \exists \text{ sphere with centre } (a, b, c)$$
 and radius r supports

$$(x - a)^2 + (y - b)^2 + (z - c)^2 > r^2$$

- The joining of valid tetrahedrons results in a surface over point cloud data.

However, for 2D or planar case, same procedure can be followed by just eliminating the given constant dimension.

5 Implementation

5.1 Developed Algorithm

Firstly, decision has to be made whether given point cloud is of two-dimensional body or three-dimensional structure, and accordingly, MCA will be selected automatically. For 3D, algorithm works as follows:

Step (1). Matrix A contains 3D point passed to DT (tetrahedron for 3D points), where circumsphere is made out of 4 points and circumradius is calculated.

Step (2). Check the valid circumspheres on matrix A and store their radius in descending order in matrix B.

Step (3). Decision is to make at this step, whether radius is selected by program or user. By default, program selects median of radius for mesh construction. But, user can give the radius input between max-radius and min-radius by user's past experience with code (Fig. 3).

Step (4). The valid tetrahedrons that satisfy the Delaunay condition are stored in a matrix C in the form of triangular faces, i.e., 1 tetrahedron = 4 triangular faces.

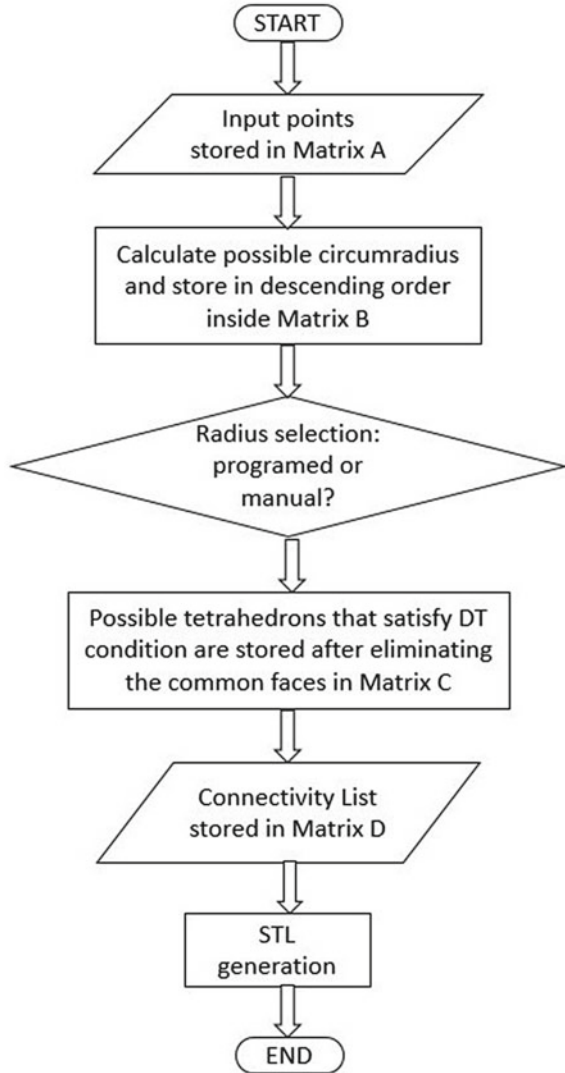
Step (5). By eliminating the common faces, a triangular surface mesh is obtained and their point to point connectivity stored in Matrix D.

Similar steps will be followed for 2D, just check the Delaunay condition using 3 points, store the satisfying points indexes in the connectivity list (refer Figs. 2 and 3).

5.2 STL Generation from Triangulated Mesh

STL is a universally accepted file format in AM for fabricating three-dimensional parts. It has become a “defacto” standard for transferring CAD data into AM software environment. Any system or software related to AM can accept STL as an input file. ASCII and Binary are two formats of STL in which a CAD data can be exported as per user requirement. In the current study, developed triangulated mesh is exported in ASCII file format. The ASCII format is selected as it is human readable. In this format, information about triangle is represented with the help of three coordinates and associated surface normal.

Fig. 2 Steps followed in developed algorithm



In first step, the output obtained from MCA, which contains connectivity list of triangular mesh, is processed to generate STL file. Let n be the number of faces created by MCA, and thus, a loop is started from $i = 1$ to n to make ASCII type STL. On termination of loop, the file is saved with .stl extension. The generated STL is then fed into the Solidworks to observe the quality (Fig. 5). An example of nose was taken and converted into STL file from obtained triangulated mesh, as shown in Fig. 4.

The discussed algorithm can be coded in any language, but MATLAB (2015 or above) provides “Alpha Shape” and “Boundary Facets” functions that ease to

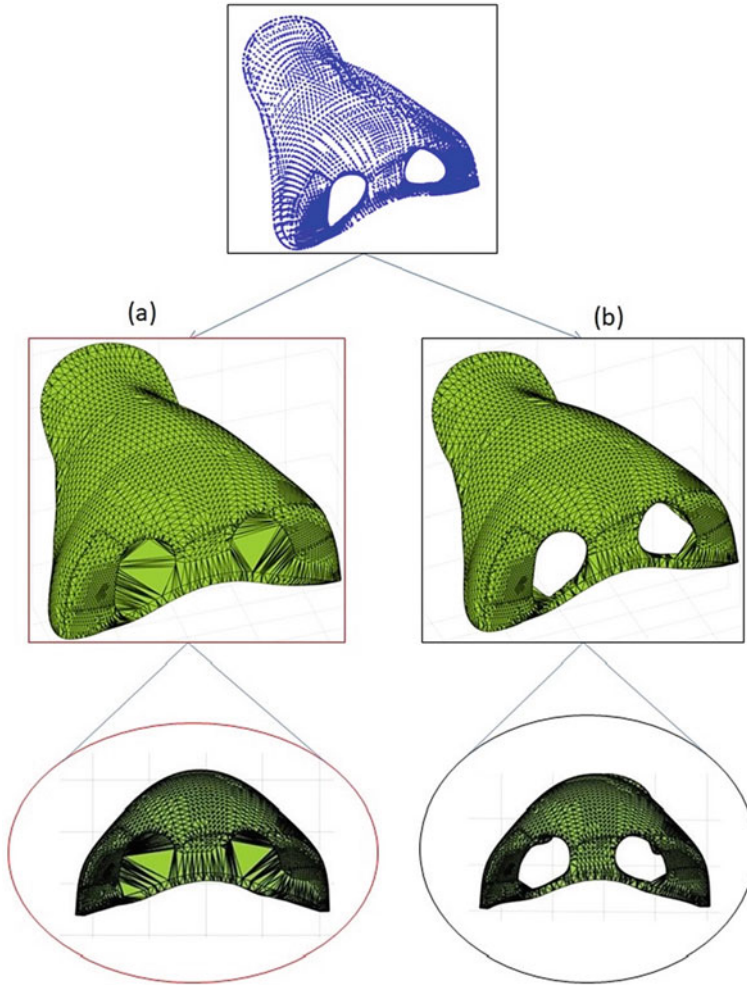


Fig. 3 Importance of radius selection in MCA: mesh created by **a** non-optimum radius, whereas **b** optimum radius

deploy the MCA. Alpha Shape creates tessellation for both 3D and 2D point cloud data. Whereas, Boundary Facets give connectivity list of points. Figure 6 shows the graphical user interface (GUI) of implementation of MCA in MATLAB. Using the developed GUI, user can interact with the point cloud data with ease and can create mesh as per the requirement.

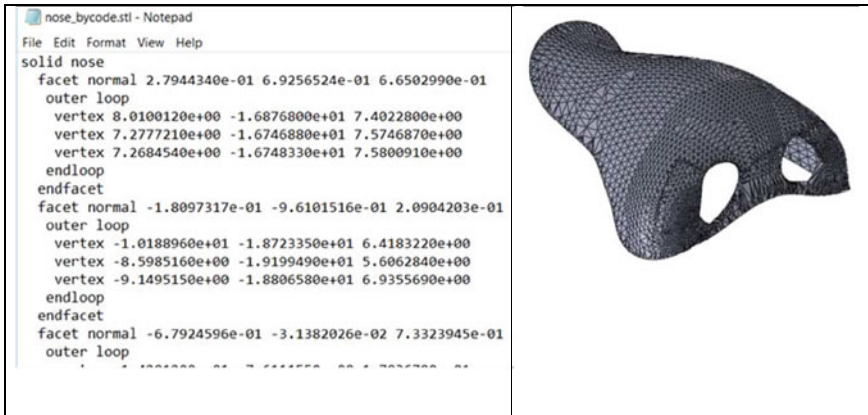


Fig. 4 Shows the generated STL file form MCA opened in (Left) Notepad, (Right) Solidworks

6 Validation

In order to show the robustness of proposed algorithm, STL file generated through MCA has been printed through fused filament fabrication (FFF) AM system. FFF offers part fabrication at relatively low cost as compared to other AM systems. In the current study, a FFF printer FlashForge creator pro along with flashprint software was used to fabricate samples. Figure 7 shows the STL file of nose exported in environment of flashprint software.

Proposed algorithm is able to generate STL for complex structures, which are validated by fabricating through 3D printing. Table 1 represents the fabricated samples and the used point cloud data. Moreover, the processing time is also presented which is measured from exporting point cloud data in MATLAB to STL generation, excluding the generating printing code and manufacturing time. Sample printing was done with acrylonitrile butadiene styrene (ABS) material by considering standard process parameters.

From the results, it can be seen that printed parts have similar structure and shape as possess in point cloud data. It means prepared STL are printable and can be printed by using any standard AM system. Overall, 3D printing results show the robustness of proposed algorithm.

7 Conclusion

In this paper, a generalized methodology of converting point cloud data into STL was presented. mesh construction algorithm (MCA) proposed solution was efficient and able to work on unorganized point cloud of complex shapes. The STL files related to architectural and biomedical sectors were successfully developed and fabricated. The

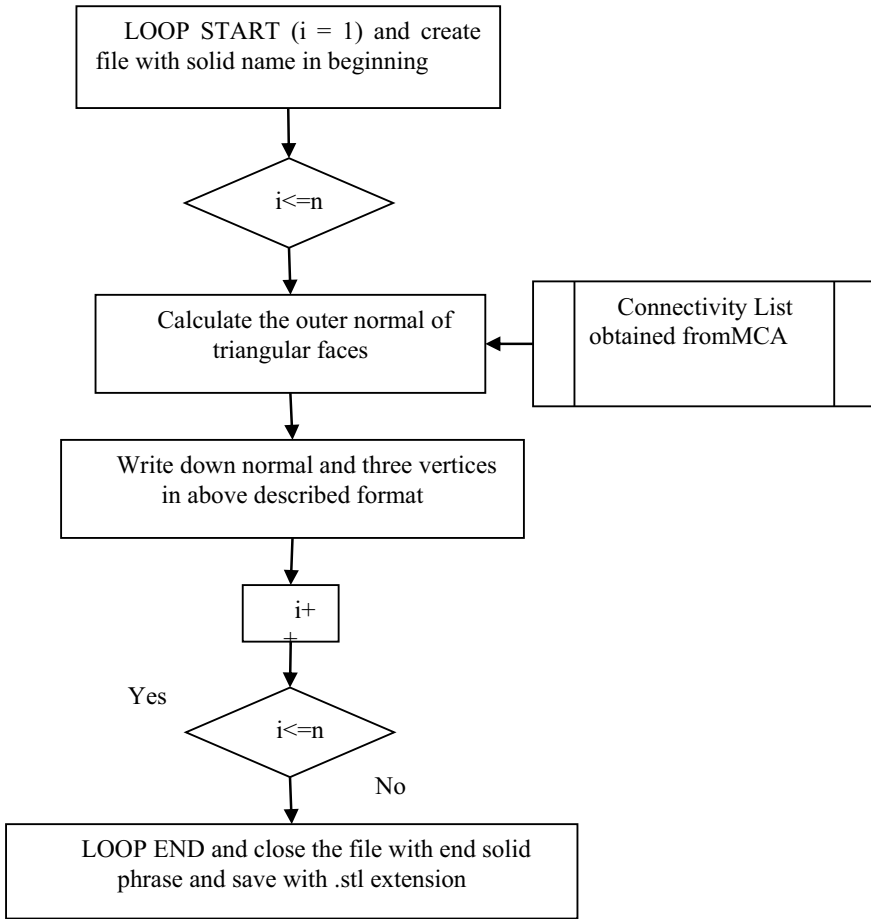


Fig. 5 Flow chart of STL file generation from Matrix D

results suggested that the dependency on software can be minimized to produce STL from point cloud. In future, the study may be extended to reduction of computational time by applying data segmentation on point cloud. Moreover, the data segmentation and filling region can be done by integrating MCA with advance algorithm of STL repairing and point reducing.

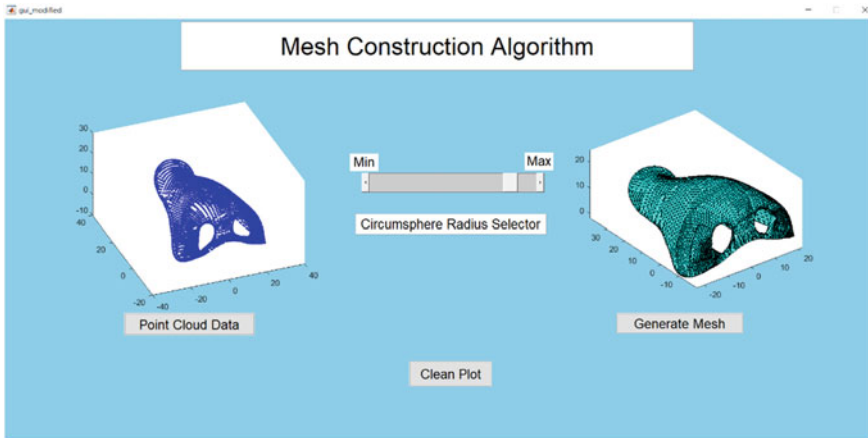




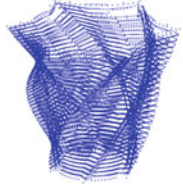





Fig. 6 Developed graphical user interface (GUI)

Table 1 Printed samples and processing time

Input Points	Part Volume (mm ³)	Point Cloud	3D printed part	Time (seconds)
206,682 (Bone)	4 × 13 × 4			619
270,036 (Human Nose)	45 × 60 × 27			812
100,926 (Pot)	45 × 45 × 40			156
151,392 (Sunglass)	32 × 17 × 10			337

References

1. Choi BK, Shin HY, Yoon YI et al (1988) Triangulation of scattered data in 3D space. *Comput Des* 20:239–248
2. Boissonnat J-D, Geiger B (1992) Three dimensional reconstruction of complex shapes base on the Delaunay triangulation. *Biomed Image Process Biomed Vis* 1905:964–975
3. Várady T, Martin RR, Cox J (1997) Reverse engineering of geometric models—an introduction. *Comput Des* 29:255–268
4. Piegl LA, Tiller W (2000) Surface approximation to scanned data. *Vis Comput* 16:386–395
5. Woo H, Kang E, Wang S et al (2002) A new segmentation method for point cloud data. *Int J Mach Tools Manuf* 42:167–178
6. Chen YH, Ng CT, Wang YZ (1999) Generation of an STL File from 3D measurement data with user-controlled data reduction. *Int J Adv Manuf Technol* 15:127–131
7. Lee SH, Kim HC, Hur SM et al (2002) STL file generation from measured point data by segmentation and Delaunay triangulation. *CAD Comput Aided Des* 34:691–704
8. Christiansen HN, Sederberg TW (1978) Conversion of complex contour line definitions into polygonal element mosaics. *Comput Graph (ACM)* 12:187–192
9. Xu J, Hou W, Zhang H (2018) An improved virtual edge approach to slicing of point cloud for additive manufacturing. *Comput Aided Des Appl* 15:330–336
10. Oropallo W, Piegl LA, Rosen P et al (2018) Point cloud slicing for 3-D printing. *Comput Aided Des Appl* 15:90–97
11. Yuan T, Peng X, Zhang D (2018) Direct rapid prototyping from point cloud data without surface reconstruction. *Comput Aided Des Appl* 15:390–398
12. Jinting Xu, Hou W, Sun Y, Lee Y-S (2018) PLSP based layered contour generation from point cloud for additive manufacturing. *Robot Comput-Integrated Manuf* 49:1–12