COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 3, 390–398
https://doi.org/10.1080/16864360.2017.1397889

Taylor & Francis
Taylor & Francis Group

Direct rapid prototyping from point cloud data without surface reconstruction

Tianyun Yuan, Xiaobo Peng and Dongdong Zhang

Prairie View A&M University, USA

ABSTRACT

To save cost and manufacturing time, prototyping technology is increasingly playing an irreplaceable role in the industry. Rapid prototyping and reverse engineering are two major technologies that meet the demands of the development. The existing approaches of directly prototyping a physical object involve various complex processing steps, i.e. reconstructing CAD model from the scanned point data, and/or STereoLithography (STL) model generation. However, such processes require professional knowledge and skills, and are laborious and far from automatic. This paper introduces a direct prototyping system, which automatically print the object from scanned point cloud. Neither a CAD model nor a STL model is created in the proposed system. In the proposed system, the two-dimensional (2D) contours in each printing plane is generated by intersecting the printing plane and Moving Least Square (MLS) surface. The proposed system simplifies the whole workflow by integrating the point-cloud projecting process, printing path generating process and 3D printing process. A user-friendly interface is developed to streamline the process.

KEYWORDS
Rapid Prototyping; Three-Dimensional (3D) Printing; Direct Manufacturing; Point Cloud; Moving Least Square (MLS) surface

1. Introduction

Rapid prototyping (RP) technology has drawn a lot of attention. The goal of rapid prototyping is to reduce the manufacturing time and be cost-effective for small batch-size production and various complex geometry designs [12]. People can manufacture their own part with a computer-aided design (CAD) model anytime and anywhere. Current technology permits the facet model, such as STL file, for printing, which can be transferred from the CAD model.

One of the most popular technologies in 3D printing is Fused Deposition Modeling (FDM). To fabricate the object, the wire-shaped material will be melted in the nozzle and deposited on the previous layer. Thanks to the fast and inexpensive manufacturing process, rapid prototyping technology has been applied in engineering industries, education, medical applications, fashion design, and even biomedical researches.

Reverse engineering (RE) refers to the process of copying or creating an exciting physical object or surface in computer environment [16]. A CAD model is built based on the points on or close to the target object or surface. The technology has been applied in auto industry, aerospace industry, medical fields, or custom product industries [17], in which it involves many freeform surface reconstructions. The cloud point data is captured using 3D sensors, such as 3D scanners, X-ray, magnetic, mechanical contact digitizing devices. The captured information usually includes the position of the points in the 3D coordinate system [15]. Meanwhile, the normal directional vector of the point also can also be obtained and output by the measuring device.

In traditional process, to rebuilt or duplicate a physical object includes several steps. First, the point cloud is collected by a 3D scanner, and then a solid CAD model has to be created based on the input cloud data. Next, the CAD model has to be tessellated into a facet model, such as STereoLithography (STL) file. Then the facet file is sliced to generate the contours on each slice so as to generate the printing paths. The entire working flow is shown in Fig. 1.

However, laborious works and professional knowledge are required in the traditional process. The users have to be familiar with various modeling software. Such skill requires users' long-years' experience in using the software. In addition, accuracy is sacrificed in both RE and RP processes. The error resulting from CAD model and faceted model is shown in Fig. 2(a). Another error is named as "stair-step" error, which is caused when printing the sliced object [4], shown in Fig. 2(b). In addition, the size of the file is concerned by some researchers [12]. STL format, a triangular facet boundary surface model,

CONTACT Tianyun Yuan tyuan@student.pvamu.edu; Xiaobo Peng xipeng@pvamu.edu; Dongdong Zhang peterzdd_2002@hotmail.com

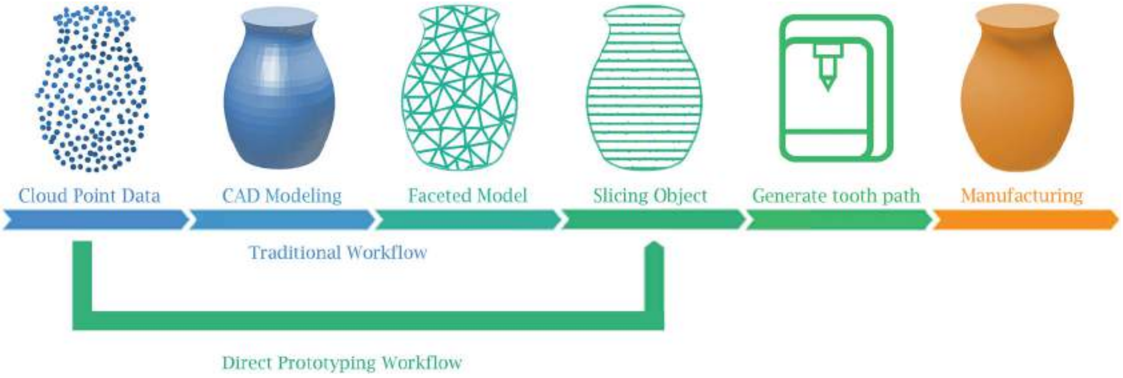


Figure 1. The comparison between traditional prototyping and proposed direct prototyping methods.

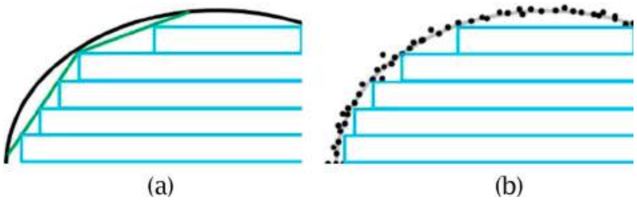


Figure 2. (a) Error in transform CAD model into STL format and the error in slicing the STL model, (b) error in direct slicing the scanned model.

provides the position of the three vertex of each triangle and the normal vector of the triangle face [15]. The more accurate the model is, the more faces are required, which requires large space to restore the information.

To address the drawbacks of the traditional processes discussed above, an automatic prototyping system is proposed in this research, which fully integrates reverse engineering and rapid prototyping process. In the proposed system, the cloud points from 3D scanner are obtained as the input, and then the point cloud will be directly sliced to generate the sliced contour with the proposed algorithm. Neither a CAD model nor a STL model is created in the proposed system. Since FDM method has been widely commercialized to the market for the education, individual design, and prototyping [11], a FDM printer will be connected in our system to follow the printing path and print the target object. The system is implemented using MATLAB, and a user-friendly interface is designed for the proposed system.

2. Literature review

Recently, there is an increasing interest in 3D printing technology, which is also named as additive manufacturing technology. Many branches of the 3D printing technology have emerged because of different material, material phases, working environment, and fabricate technology [11]. Meanwhile, 3D scanning and surface reconstruction technologies are developing rapidly with a growing demand in the market.

2.1. Researches in prototyping manufacturing process

The goal of prototyping is to reduce manufacturing time and material cost, as well as to improve the printing quality. Many aspects of 3D printing have been discussed by researchers, including printing orientation [15], slicing methods [13] (both slicing from polygon surface models or direct slicing from CAD model), printing materials [11], supporting structures [14] (the ways to reduce the cost on the supporting parts and different structures of supporting), printing pattern [8],[15] (the efficiency of printing solid model and different printing path), etc. However, among these studies, most of the methods are based on the STL format or CAD model.

In all the technical aspects of 3D printing, slicing is the most necessary process which cannot be avoid in any geometry or manufacturing method. The object is sliced into several planes and fabricated layer by layer. A sliced file is required to control the printer. There are three approaches to create the sliced file. (1) Most approaches are to generate the sliced file from the faced model, such as .stl model and .obj model. (2) Some studies have been done to directly compute the path from CAD model, which will reduce the error caused by tessellation. (3) Another group of people have researched on generating the sliced file directly from cloud point data. A number of works have been done to optimize the slicing process.

2.2. Researches in model reconstruction from point cloud data

With the development of three dimension scanning technology, reverse engineering has been studied and applied in many research areas and cases. The goal of reverse engineering is to represent the object in CAD model or faceted model from cloud point data. Lin et al. [10] led his team to build a customized joint using the point cloud data scanned from the patients and find more suitable prosthesis. Bey et al. [2] presented the reconstruction of 3D models from point cloud data acquired in industrial environments using priori CAD models. The reconstruction technology has also been applied for archaeology and cultural heritage purpose [7], representing a reality-based environment of ruined archeological.

Reconstructing the object from cloud points can be considered as the work related to probability and statistics. The raw scanned points include unpredictable noise points. Various mathematical methods have been studied to find the representative points, thus to reduce mass data into smaller amount of points. In this process, how to better represent the surface and geometry with high accuracy and less calculation time are mainly considered. Fabio [6] has discussed and introduced various surface representation methods. He classified different surface reconstruction algorithms. Polygonal surface has been studied a lot and triangulation has been discussed most.

For 3D printing, a sliced model is required for manufacturing an object. Thus, directly generating the sliced model from point cloud is becoming promising topic. Instead of finding the 3D vertex, the alternative is to determine the surface points of the 2D slicing contour. Iterative closest point (ICP) method has been discussed by Bouaziz et al. [3]. The key of this method is to find the corresponding points between two point sets, and to calculate the conversion coordinates from the original coordinates based on the input point data. However, ICP method is sensitive to the initial points and noisy points. Moving least Square (MLS) method has been studied in statistics area and has been applied in several mathematical models. MLS surface method is to use MLS method for the surface representation based on the geometric information of the original point cloud. Levin initially defined the MLS surface in paper [9]. In the MLS method, an energy function is introduced to calculate search for the surface points by considering the local minimum energy [1], [9]. Dey and Sun [5] improved the algorithm by incorporating the size of the local features. An image-based reduction method is proposed by Liu et al. [12]. They calculated average projection error to determine the sections of the point data. If the error is larger than the tolerance, the section will be divided into two sections. The feature points will be kept in each region within the user-defined tolerance. Layer contours will be generated based on the feature points, and then faired with a discrete curvature. Besides, the concept of adaptive slicing was applied by Wu et al. [16]. They applied the concept in probability theory to determine the step size in reconstruction process. They calculated the correlation coefficient of the project neighbor area to determine the step length, so as to keep sharp features in the contour. In paper [17] and [18], curvature and the radius of curvature were calculated from the point cloud to guide the manufacturing process to achieve the best production efficiency.

3. Methodology

In the proposed direct prototyping system, 1) the scanned point cloud is processed using Moving Least Square (MLS) surface representation, 2) the input data is analyzed and divided into slabs and groups, 3) the slicing contour is generated in each slicing plane using MLS method, and 4) an integrated system is developed to achieve the direct printing automatically. The workflow is shown in Fig. 3. The raw point cloud is processed using the MLS method, so the noisy data is handled automatically and the geometric information of the point cloud is obtained. With the MLS surface representation, the slicing contour on each slicing plane is traced by finding the intersecting contours between the point cloud and the slicing plane. In the proposed system, both CAD model reconstruction and STL model regeneration are avoided, thus the production time is shortened. With the slicing contour on each slicing plane, numerical control (NC) G codes are generated to control the 3D printer.

3.1. *Surface representation using moving least square method*

MLS method was initially defined by Levin [9]. Amenta and Dey gave an explicit definition and implemented to surface reconstruction application later [1],[5]. Instead of simply projecting points onto the layers, the minimum local energy along the moving direction is determined. A surface point is calculated with MLS, which is used to represent the surface geometry. The process is briefly introduced in this section.

The input data is a set of points, which include the information of their position and normal. First, a guess point x is assumed, shown as the square dot in Fig. 4 (a). A set of neighbor points p_i (shown as the blue dots in Fig. 4.) is selected by calculating the distance between x and p_i. In this research, the 30 most closest points are selected as neighbor points. A vector $\vec{\mathrm{v}_\mathrm{i}}$ is assigned as $\vec{\mathrm{v}_\mathrm{i}} = (p_i - x)$. The Gaussian weighting function is defined as:

$$\mathrm{w}(x, p_i) = e^{-\|\vec{v_i}\|^2/\mathrm{h}^2} \tag{1}$$

in which is a scale factor, determines the width of the Gaussian kernel [16]. Here it is taken as a fraction of the local feature size, $\mathrm{h}^2 = \|v_i\|^2_{\max}$. Therefore, farther

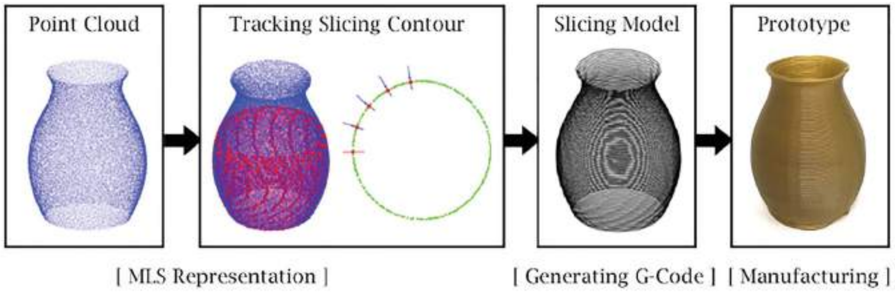


Figure 3. The workflow of proposed direct prototyping.

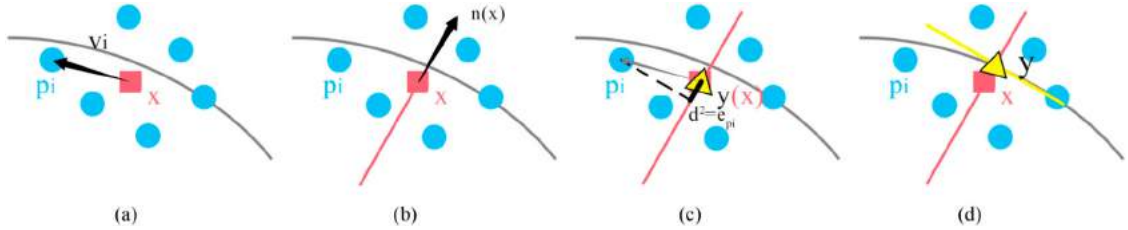


Figure 4. MLS surface representation: project the scanned point along the normal direction to achieve the local minimum energy.

points have less weight, while closer points have more weight. With the determination of the neighbor points and weighting function, the moving direction $\overrightarrow{\mathrm{n}(x)}$ is obtained by Eqn. (2). In Eqn. (2), n_{p_i} is the normal at point p_i, which can be obtained from the input point cloud data. The moving direction, shown as Fig. 4(b)., is a weighted vector. The point on the MLS surface will be found in the moving direction.

$$\overrightarrow{\mathrm{n}(x)} = \frac{\sum_{p_i \in P} w(x, p_i) \times \overrightarrow{n_{p_i}}}{\left\| \sum_{p_i \in P} w(x, p_i) \times \overrightarrow{n_{p_i}} \right\|} \tag{2}$$

An estimated point y, can be found along the moving direction $\overrightarrow{\mathrm{n}(x)}$. The local energy refers to the weighted sum of the distance square between p_i and estimated point y along the moving direction, as shown in Fig. 4(c). It can be describe as following:

$$\mathrm{e}(y, \overrightarrow{n(x)}) = \sum_{p_i \in P} \left((y - p_i) \cdot \overrightarrow{n(x)}\right)^2 \cdot w(y, p_i) \tag{3}$$

Where y can be substituted as $\mathrm{y} = x + t \times \overrightarrow{n(x)}$. t is the moving distance between guess point x and the estimated point y along $\overrightarrow{\mathrm{n}(x)}$. The energy function can be restated as a function of t:

$$\mathrm{e(t)} = \sum_{p_i \in P} (((x + t \cdot \overrightarrow{n(x)})) - p)) \cdot \overrightarrow{n(x)})^2 \cdot w(y, p_i) \tag{4}$$

The surface point is the estimated point, whose $\mathrm{e}(t)$ reaches to the smallest energy. It means the final moving distance is determined by finding the minimum local energy along the moving direction $\overrightarrow{\mathrm{n}(x)}$. Then the guess point x is projected to the surface point. The next guess point is calculated by add a step size in the direction tangent to the normal of previous surface point until a closed contour is formed. After finding all the surface point on the slicing plane, they will be connected as a slicing contour.

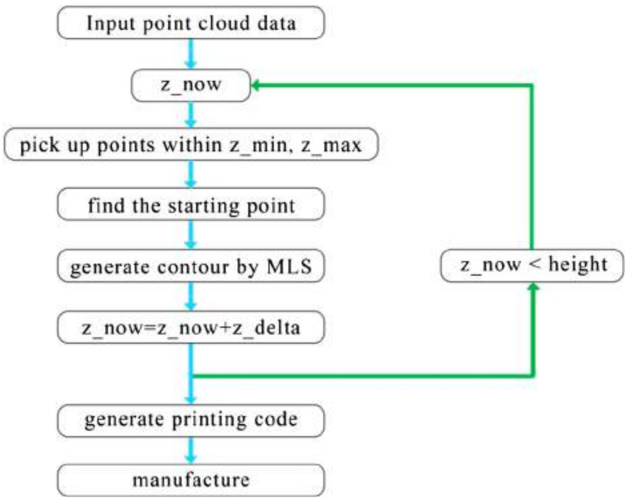
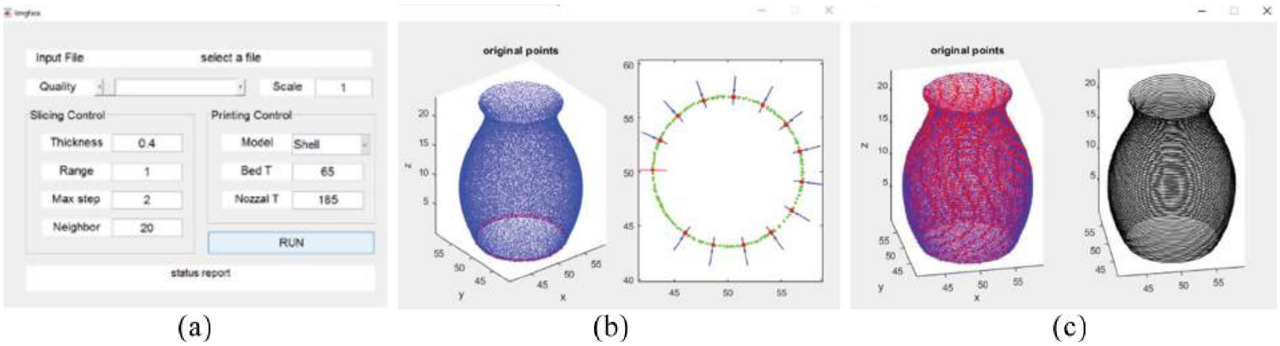


Figure 5. Detail process in algorithm.

3.2. Tracing the slicing contour on each slicing plane

The building direction is assumed in z axis. The process of the algorithm is presented in Fig. 5. The details are explained below:

Step 1: Input point generation. The input data is point cloud data obtained from a 3D scanner. The input



Input File select a file

Quality Scale 1

Slicing Control

Thickness 0.4

Range 1

Max step 2

Neighbor 20

Printing Control

Model Shell

Bed T 65

Nozzal T 185

status report

Figure 6. Designed interface: (a) Processing parameters, (b) 2D contour tracing in each slicing plane, (c) 3D printing data generation.

includes six sets of data, representing the x, y, and z coordinates of the scanned points and the normal vector i, j and k of the scanned points.

Step 2: Point selection on each slicing layer. Starting from the lowest height of the input data, a bounding box is set by presetting the range (height) in z axis. Only the points enclosed in the bounding box are used in the calculation described in Step 3. This will save calculation time. If there is multiple closed contours appear in one layer, these points will be divided into groups and generate the contours separately.

Step 3: Contour generation. The surface points are traced using the MLS method. In each layer, the starting point is selected in the proposed algorithm. The next guess point is calculated by add a step size in the direction tangent to the normal of previous surface point until a closed contour is formed. After re-arranging the surface points, the contour is the linear connection of these traced points in sequence.

Step 4: Layer-by-layer contour generation. Moving to the next layer, the slicing contour will be traced by repeating Steps 2 and 3 until printing height (z_now) reaches the maximum height of the input data.

Step 5: Product manufacturing. All contour data is sent to the 3D printer to manufacture the product.

3.3. Designed interface

An interface is designed using MATLAB as shown in Fig. 6. The input data is shown in the left plotting area, and the slicing contour is traced in the right plotting area as shown in Fig. 6(b). The final work is shown in Fig. 6(c). There are two sets of parameters on the interface, including “slicing control” parameters and “printing control” parameters. Slicing control parameters include “Thickness”, “Range”, “Max step”, and “Neighbor”. Printing control parameters include “bed temperature” and “nozzle temperature”. “Thickness” is the height of each layer (z_delta as in Fig. 5). It usually ranges from 0.1 mm to 0.4 mm in most commercial printers. The smaller the layer thickness is, the smoother the surface is. “Range” is the height of the bounding box to select points on each layer, which is recommended as 0.6–3 mm. “Max step” is the distance between the surface point and the next guess point. “Neighbor” is the number of points used to calculate the surface points in MLS surface. “Bed temperature” and “Nozzle temperature” are the temperatures of the workbench and the extruder of 3D printer.

Table 1. Bed temperature and nozzle temperature of different printing materials.

Material	Bed Temperature °C	Nozzle Temperature °C
PLA	60–80	180–200
ABS	80–100	210–240
Ninja Flex	20–40	220–250

The printing parameters are determined according to the printing materials. The bed temperature and nozzle temperature are two main parameters influencing the printing quality. Tab 1 shows the temperatures of some materials that has been tested with our printer.

With one click on the “RUN” button, the point cloud is imported and the MLS algorithm is initiated. Then the 2D contours are generated, as shown in Fig. 6 (b) and (c). Finally, the final product will be printed out automatically.

4. Results and discussion

4.1. Slicing contour tracing

The quality of the slicing contour is affected by the height of bounding box, step length, neighbor point size and the quality of input files. The first three parameters can be controlled in our system. A cylinder model is used as the sample to test how the parameters affect the accuracy of the sliced contour. In the analysis, the error is defined as the difference of cylinder radius and the distance of the surface point to the cylinder center. The analysis is on the 2D slicing plane, using x and y coordinates. In Fig. 7, Fig. 9, and Fig. 11., the blue dots are selected as input

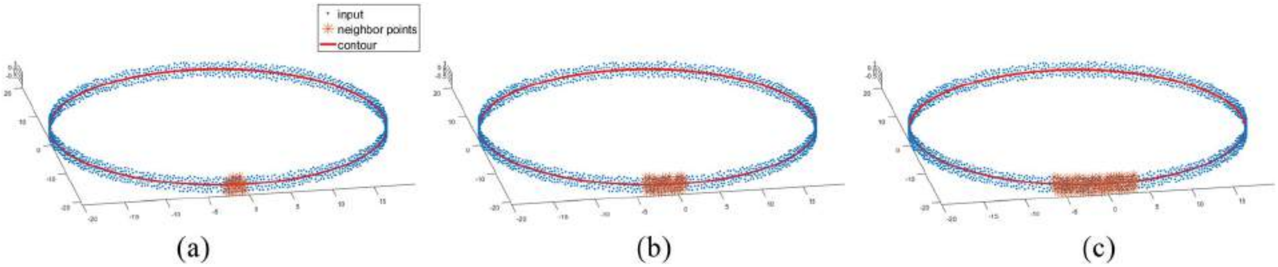


Figure 7. Neighbor size influence: (a) neighbor size is 30 points, (b) neighbor size is 60 points, (c) neighbor size is 120 points.

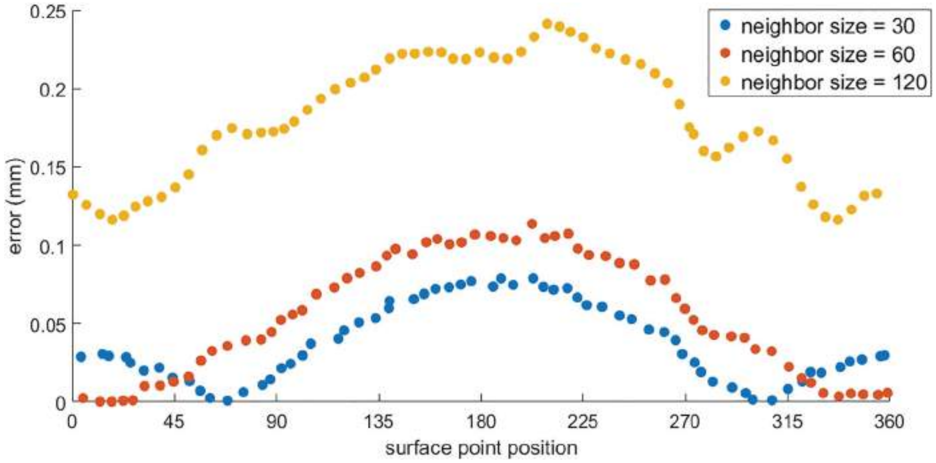


Figure 8. Error analysis related to the neighbor size.

points in the bounding box. The red line is the contour generated by using MLS method. The orange star dots are the neighbor points for the first guess point. The neighbor points are closest points to the guess point selected by calculating the distance between guess point and all the points in the bounding box. The red star dots in Fig. 11 are the surface points.

Neighbor size determines the number of the sample points selected to compute the surface point, which influences accuracy of the contours. The height of bounding box in Fig. 7 is 2.0 mm. The numbers of neighbor point are 30 points (in Fig. 7(a).), 60 points (in Fig. 7(b).), and 120 points (in Fig. 7(c).). When the neighbor size is too big, the sample of neighbor points for MLS computation spreads wide and leads to low accuracy, as shown in Fig. 7(c). Fig. 8 shows the error of each surface point on one contour with respect to different sample size. The results show that neighbor size 120 has the biggest error while neighbor size 30 has the smallest error.

The height of bounding box can neither be too big nor too small. Fig. 9. shows how the height affects the results of contour tracing. The heights are 0.5 mm, 1.0 mm, and 4.0 mm respectively in Fig. 9. It is worth to note that the computation time will increase when the height of bounding box is bigger. Fig. 10(a). shows the error of each surface point on one contour with respect to different bounding box height. It indicates that 0.5 mm bounding box height causes the biggest error while 4 mm bounding box height has the smallest error. Fig. 10(b) shows the mean error of all the surface points on one contour with respect to different bounding box height. It reveals that when the bounding box height is bigger than 3.5, the mean error is the same. Thus, to save the calculation time, using the bounding box 3.5 mm is good enough in this example.

Step length is another factor contributing to the accuracy and computation time. Step length is how far to move from the current surface point to the next guess point along the perpendicular vector of the surface point normal. Fig. 11 shows the results with different step length as 0.5 mm, 2 mm and 5 mm. The projection area in Fig. 11 is 40 mm×40 mm. Small step length, as shown in Fig. 11(a)., results in more iterations and increases calculation time. The contour on the slicing plane is represented by 252 line segments. When the step length is 5 mm as in shown Fig. 11(c), only 27segements is generated to represent the contour. The error is bigger than Fig. 11(a) and Fig. 11(b).

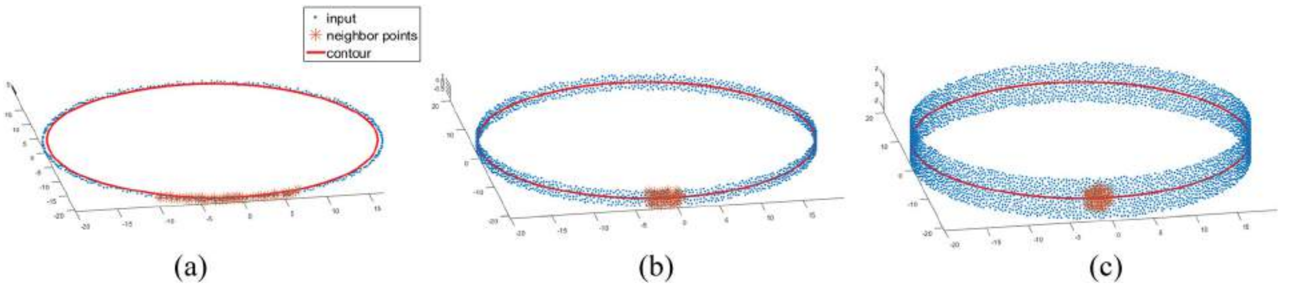


Figure 9. The height of bounding box influence: (a) 0.5 mm height, (b) 1.0 mm height, (c) 4.0 mm height.

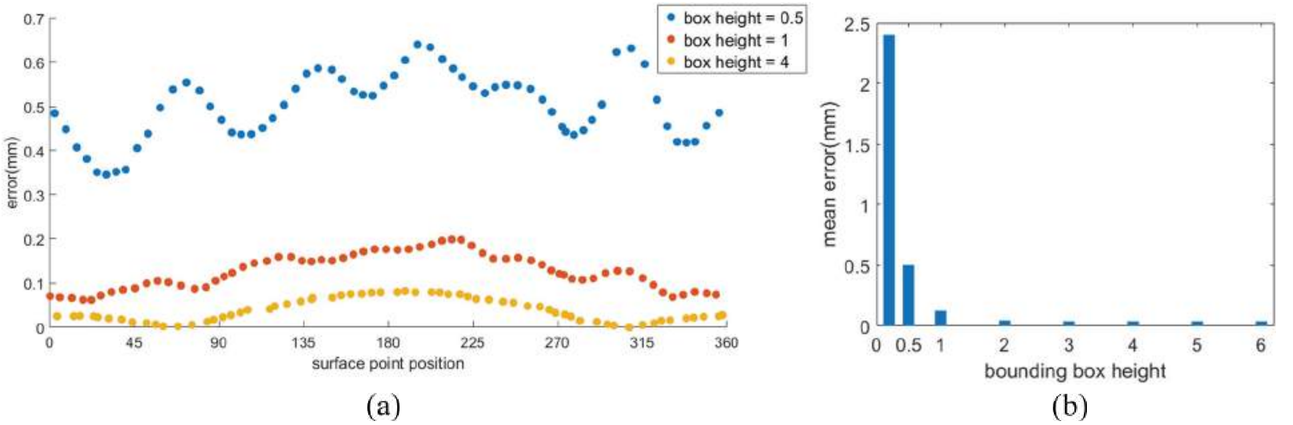


Figure 10. Error analysis related the height of bounding box: (a) the error of each surface point, (b) the relation between the height of bounding box and its mean error.

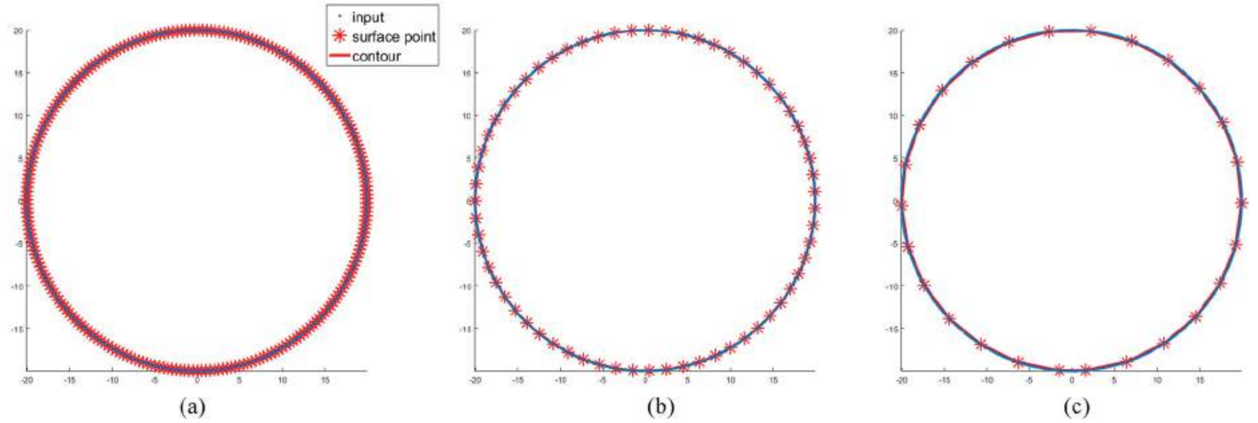


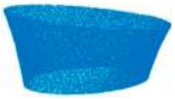
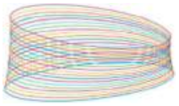
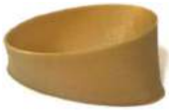
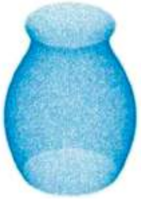
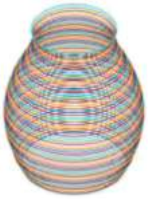
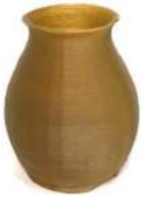
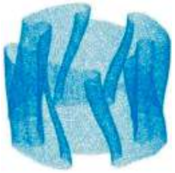
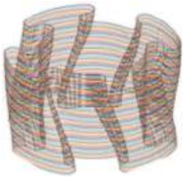
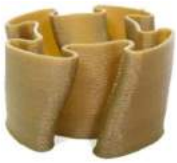
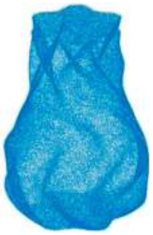
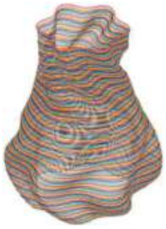
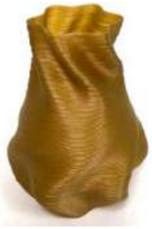
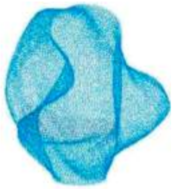
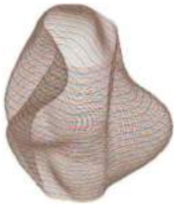
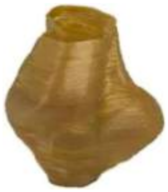
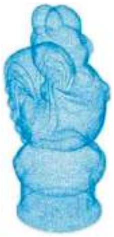
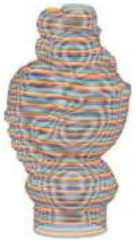
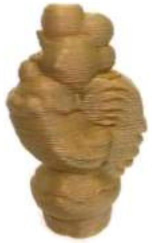
Figure 11. Step length influence: (a) step size is 0.5 mm, (b) step size is 2 mm, (c) step size is 5 mm.

4.2. 3D printing

The examples printed by using the proposed system are presented in Tab. 2. The size of these samples is within 80 mm by 80 mm by 80 mm. The processing time using the proposed system is presented in Tab. 2 as Time. The processing times are measured from importing the input file to generating the printing code, excluding the manufacturing time.

As shown in Tab. 2, the processing times range from 129 second to 402 second. With the traditional process, a CAD model needs to be created from point cloud data using software, such as ImageWare, or X design. Then, the CAD file will be refined and converted into STL model using modeling software, such as NX Siemens, or SolidWorks. Slicing software, like Cura, Slic3r, is required to slice the STL model and export G code for 3D printing. Such processes require professional knowledge and skills. Compared to the traditional process, the processing time using the direct 3D printing system is dramatically reduced.

Table 2. Printed samples and processing times.

	Input points	Object size(mm)	Input data	Sliced object	Prototypes	Time (Second)
A	12,519	20×20×8				139.9
B	13,975	18×18×23				127.2
C	23,567	36×36×20				318.3
d	29,131	45×45×60				181.5
e	14,516	47×43×61				401.8
f	21,534	22×32×60				164.7

5. Conclusions

In this paper, a direct rapid prototyping system is designed. The direct rapid prototyping system greatly simplifies the process by avoiding troublesome file conversion between different software, and complex professional modeling steps. With the proposed system, it can well reconstruct the object with freeform surface which are hard to model. The time consumed in the whole workflow has been greatly reduced.

Currently the system can only print the surface of the 3D object. Future work will be focused on generating filling path inside the contours in order to print solid object. In addition, future work is to develop algorithm to enable the generation of multiple contours in each sliced layer.

ORCID

Tianyun Yuan http://orcid.org/0000-0001-6846-6550
Xiaobo Peng http://orcid.org/0000-0002-0498-7194
Dongdong Zhang http://orcid.org/0000-0001-9471-1628

References

[1] Amenta, N.; Kil, Y-J.: Defining point-set surfaces, ACM Transactions on Graphics (TOG), 23(3), 2004, 264–270. https://doi.org/10.1145/1015706.1015713

[2] Bey, A.; Chaine, R.; Marc, R.; Thibault, G.; Akkouche, S.: Reconstruction of consistent 3D CAD models from point cloud data using a priori CAD models, ISPRS Workshop on Laser Scanning, 2011, 1.

[3] Bouaziz, S.; Tagliasacchi, A.; Pauly, M.: Sparse iterative closest point, Computer Graphics Forum, Blackwell Publishing Ltd, 32(5), 2013, 113–123. https://doi.org/10.1111/cgf.12178

[4] Choi, S.-H.; Samavedam, S.: Modelling and optimisation of rapid prototyping, Computers in Industry, 47(1), 2002, 39–53. http://doi.org/10.1016/S0166-3615(01)00140-3

[5] Dey, T.- K.; Sun, J.: An adaptive MLS surface for reconstruction with guarantees, in the Proceedings of the Third Eurographics Symposium on Geometry processing, 2005, 43–52.

[6] Fabio, R.: From point cloud to surface: the modeling and visualization problem, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 34(5), 2003, W10. https://doi.org/10.3929/ethz-a-004655782

[7] Guidi, G.; Russo, M.; Angheleddu, D.: 3D survey and virtual reconstruction of archeological sites, Digital Applications in Archaeology and Cultural Heritage, 1(2), 2014, 55–69. https://doi.org/10.1016/j.daach.2014.01.001

[8] Jin, Y.; He, Y., Xue, G.-H.; Fu, J.-Z.: A parallel-based path generation method for fused deposition modeling, The International Journal of Advanced Manufacturing Technology, 77(5–8), 2015, 927–937. https://doi.org/10.1007/s00170-014-6530-z

[9] Levin, D.: The approximation power of moving least-squares, Mathematics of Computation of the American Mathematical Society, 67(224), 1998, 1517–1531. https://doi.org/10.1090/S0025-5718-98-00974-0

[10] Lin, Y.-P.; Wang, C.-T.; Dai, K.-R.: Reverse engineering in CAD model reconstruction of customized artificial joint, Medical Engineering & Physics, 27(2), 2005, 189–193. https://doi.org/10.1016/j.medengphy.2004.09.006

[11] Lipson, H.; Kurman, M.: Fabricated: The New World of 3D Printing, John Wiley & Sons, Indianapolis, IN, 2013.

[12] Liu, G.-H.; Wong, Y.-S.; Zhang, Y.-F.; Loh, H.-T.: Error-based segmentation of cloud data for direct rapid prototyping, Computer-Aided Design, 35(7), 2003, 633–645. http://doi.org/10.1016/S0010-4485(02)00087-8

[13] Mohan, P.-P.; Venkata, R.-N,; Dhande S.-G.: Slicing procedures in layered manufacturing: a review, Rapid Prototyping Journal, 9(5), 2003, 274–288. http://doi.org/10.1108/13552540310502185

[14] Vanek, J.; Galicia, J.-A.-G.; Benes, B.: Clever support: efficient support structure generation for digital fabrication, Computer Graphics Forum, 33(5), 2014, 117–125. https://doi.org/10.1111/cgf.12437

[15] Venuvinod, P.-K.; Ma, W.: Rapid Prototyping: Laser-based and Other Technologies, Springer Science & Business Media, New York, 2004. https://doi.org/10.1007/978-1-4757-6361-4

[16] Wu, Y.-F.; Wong, Y.-S.; Loh, H.-T.; Zhang, Y.-F.: Modeling cloud data using an adaptive slicing approach, Computer-Aided Design, 36(3), 2004, 231–240. http://doi.org/10.1016/S0010-4485(03)00097-6

[17] Yang, P.; Qian, X.: Adaptive slicing of moving least squares surfaces: toward direct manufacturing of point set surfaces, Journal of Computing and Information Science in Engineering, 8(3), 2008, 031003. https://doi.org/10.1115/1.2955481

[18] Zhang, D.; Yang, P.; Qian, X.: Adaptive NC path generation from massive point data with bounded error, Journal of Manufacturing Science & Engineering, 131(1), 2009, 741–751. https://doi.org/10.1115/1.3010710