Title:

Direct Rapid Prototyping from Point Cloud Data without Surface Reconstruction

Authors:
Tianyun Yuan, tyuan@student.pvamu,edu, Prairie View Texas A&M University
Xiaobo Peng, xipeng@pvamu.edu, Prairie View Texas A&M University
Dongdong Zhang, peterzdd_2002@hotmail.com

Keywords:
Rapid Prototyping, Three-Dimensional Printing, Direct Manufacturing, Point Cloud Data

DOI: 10.14733/cadconfP.2017.27-31

Introduction:

With the development of computer-aided design and additive manufacturing technology, rapid prototyping is getting more and more important. Direct prototyping is an integration of reverse engineering (RE) process and rapid prototyping (RP) process [5], which converts the scanned point cloud into facet files, known as STereoLithography (STL) files, and prints out the prototype. RE process and RP process have been widely applied in automobile design, aerospace design, biological field, etc. [4]. With the increasing market of 3D scanners and 3D printers, the technology is not only applied to professional areas but also to daily lives. The traditional approach involves tedious and time-consuming processing steps, i.e., reconstructing point cloud data into CAD models or sculptured surfaces, creating STL models based on the 3D models, and finally importing STL models into commercial 3D printer, where the STL models will be sliced for printing, as seen in Fig. 1. Such processes require certain professional knowledge and skills to process point cloud, reconstruct CAD model and generate STL file. It is laborious and far from automatic, and those who have no background of RE knowledge and CAD skills can barely produce a good prototype with the point cloud. It would be more beneficial to develop a direct prototyping system which simplifies the whole procedure by receiving the raw point cloud and automatically printing the point cloud. Many research works have discussed about slicing the object from CAD model or a facet model [6]. Another approach is to directly slice the object from point cloud data [5], [7], [8].

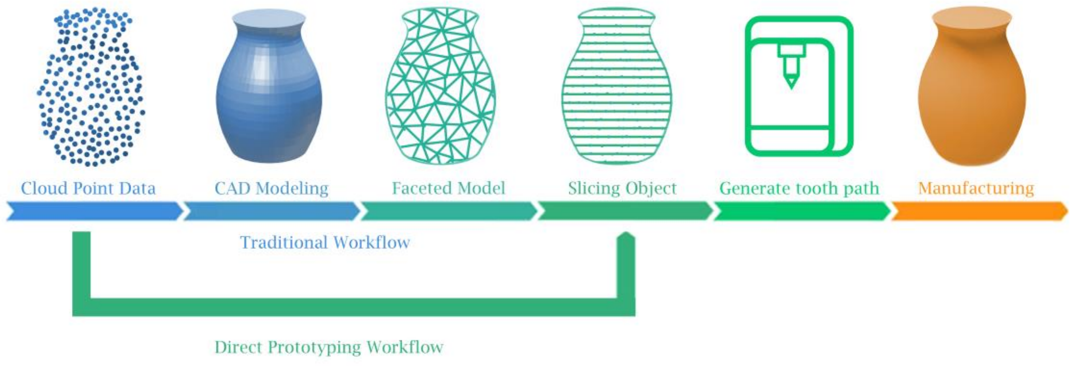


Fig. 1: The comparison between traditional prototyping and proposed direct prototyping methods.

To enable more people to experience the convenience of direct prototyping technology, we designed an automatic prototyping system, as seen in Fig. 1. In the proposed system, the point cloud will be processed and sliced automatically without the traditional CAD model reconstruction and STL model generation, which shortens the processing time.

Main Idea:
In the direct prototyping system, 1) the scanned point cloud is processed using Moving Least Square (MLS) method [1-3], 2) the sliced contour is generated in each slicing plane using MLS method [8],[9], and 3) an integration system is developed to achieve the direct printing automatically. The workflow is shown in Fig. 2. 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 defined printing thickness, multiple slicing planes/layers are used to slice the point cloud. With the MLS surface representation, the sliced contour on each slicing plane is traced by finding the intersecting curves 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 sliced contour on each slicing plane, numerical control (NC) G codes are generated to control the 3D printer. A user-friendly interface is designed to integrate the whole process, including importing the point cloud, generating the sliced contour, and sending the control signals to the 3D printer. Parameters affecting the slicing results are studied and discussed. The proposed system is tested with a FDM (Fused Deposit Modeling) printer. Examples printed by using our system is presented, and the results are compared with traditional process.

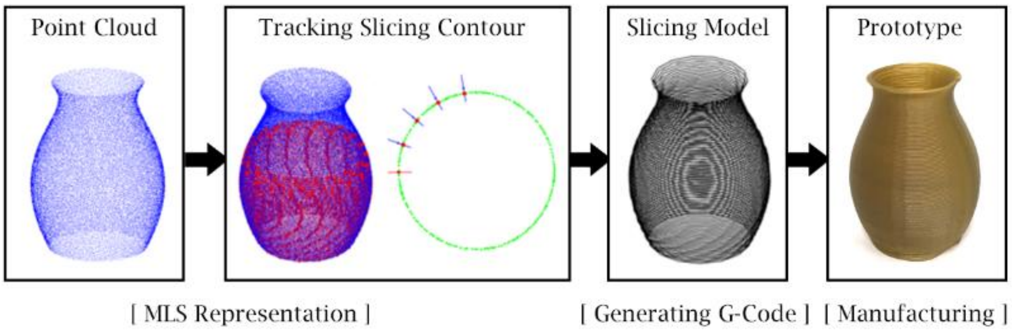


Fig. 2: The workflow of proposed direct prototyping.

Moving least square surface representation
Moving least square method was initially defined by Levin [3], and an explicit definition was given by Amenta and Kil [1]. A minimum local energy along the moving direction is calculated to determine the surface points.

A scanned point x, as the red dot in Fig. 3(a)., among a point cloud is selected. A set of input data P_i, as the blue dots in Fig. 3. is selected to calculate the distance between x and P_i. A vector v_i is calculated to each point as $v_i = (p_i - x)$, and a Gaussian weighting function is defined as:

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

in which $\| v_i \|^2_{\max}$ determine the width of the Gaussian kernel [2], as a fraction of the local feature size. Therefore, farther points have less weight, while closer points have more weight in the weighted function. With the determination of the neighbor points and weighted function, a weighted normal vector $n(x)$, in Fig. 3(b). of the neighbor points can be obtained with the normal of input data n_{p_i}:

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

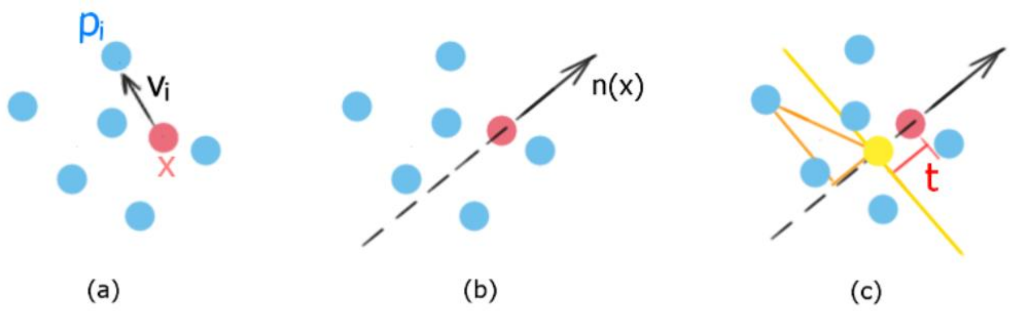


Fig. 3: MLS surface representation: project the scanned point along the normal direction to achieve the local minimum energy.

A MLS surface point, as the yellow point in Fig. 3(c)., can be found along the moving direction as $\mathrm{n}(x)$. The moving distance is determined by finding the local minimum local energy along the normal direction $\mathrm{n}(x)$. The local energy refers to the weighted sum of the distance between P_i and MLS surface point, as the yellow point in Fig. 3(c)., along the moving direction, and defined as:

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

Where y can be substituted as $\mathrm{y} = x + t * n(x)$, and the energy function can be restated as a function of t, which is the moving distance between point x and the surface point along the normal direction:

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

When $\mathrm{e}(t)$ reach to the smallest energy, the initial point x is projected to the MLS surface.

Tracing the sliced contour on each slicing plane

The 3D printing is performed layer by layer. On each layer, the two-dimensional (2D) contour is required for printing. In the proposed system, the sliced contour is traced by intersecting the slicing plane with the MLS surface. It is assumed that the building direction is z axis, the same as the direction of scanning the object. Detail of the algorithm process is shown in Fig. 4.

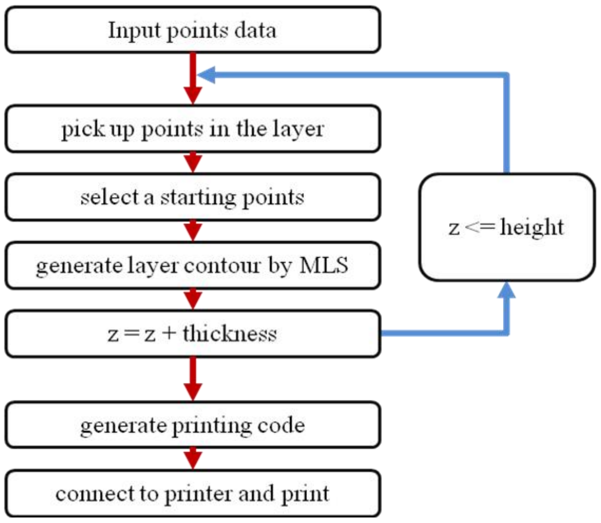
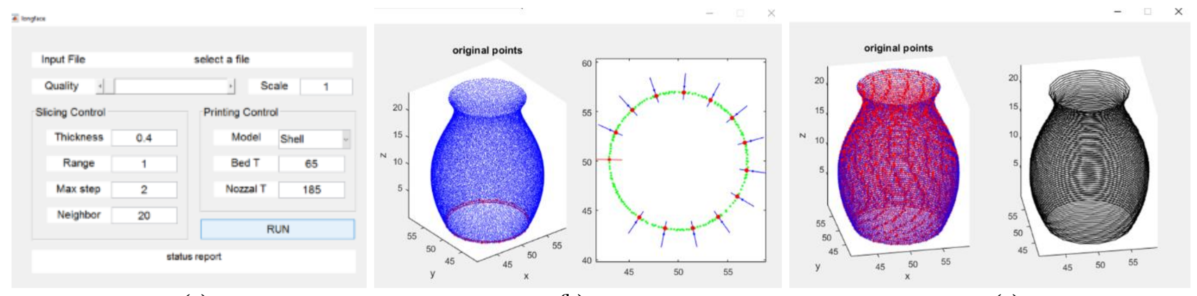


Fig. 4: Process of slicing the contours.

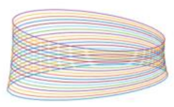
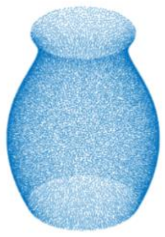
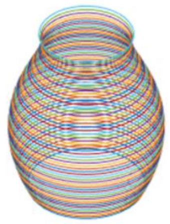
Integration interface
The whole process is integrated as seen in Fig. 5. Related parameters are set initially, such as projection thickness, printing thickness and printing temperature. With one click on the "RUN" button, the process starts with the importing point cloud and then generating 2D contours, as seen in Fig. 5(b).,(c).

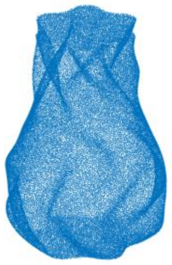
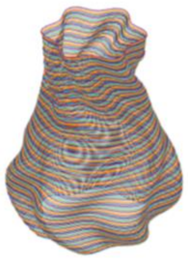
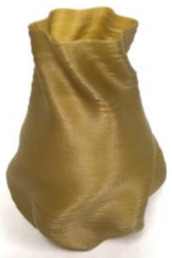
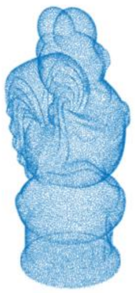
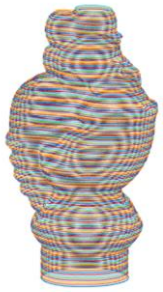
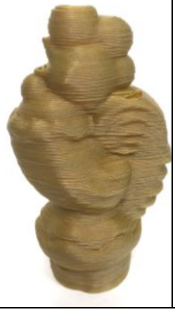

Input File: select a file
Quality
Scale: 1
Slicing Control
Thickness: 0.4
Range: 1
Max step: 2
Neighbor: 20
Printing Control
Model: Shell
Bed T: 65
Nozzal T: 185
RUN
status report

(a) (b) (c)
Fig. 5: Designed Interface: (a) Processing parameters, (b) 2D contour tracing, (c) Finished sliced object.

In the system, some processing parameters have great impact on the printing process. "Thickness" determines the height of each layer, controlling the influence of stair-effect of printed part. Different material has its own working thickness range. In all examples, we set it between 0.3 and 0.4mm, since we mainly work on PLA material. "Range" means the height of the bounding box. "Neighbor" is the amount size of sample point set P in the algorithm. "Bed T" and "Nozzle T" are the temperature of the working plane and nozzle. The temperature varies with the material. PLA requires the bed temperature and nozzle temperature around 65°C and 180°C.

Results and discussion:
Some examples printed by using the proposed system are presented in Tab. 1. The size of the samples is within 10 by 10 by 10 cm. The comparison of the processing time between our system and traditional method is presented in Tab. 1. The processing time of our system is measured from importing the input file until generating the printing code. The traditional processing time includes inputting the point cloud data, creating the CAD model and STL model, and slicing the object. However, the time of traditional method depends on individual skill level of the users and the software applied in the process. From Tab. 1., the processing time with proposed system has been greatly reduced.

	Input points	Object size(mm)	Input data	Sliced object	Prototypes	Time (s)	Traditional time (s)
a	12,519	20x20x8				139.9	341
b	13,975	18x18x23				127.2	388

c	29,131	45x45x60				131.5	1,893
d	21,534	22x32x60				164.7	>3,000

Tab. 1: Printed samples and processing times.

Conclusion:
In this paper, we present a direct rapid prototyping system fully integrating reverse engineering and rapid prototyping processes. The system automatically manufactures the object from point cloud data, without the CAD model reconstruction. No professional skills are required for the user to use the system. Compared to the traditional process, the proposed system takes less time to manufacture the product. Future work will focus on improving the system with filling patterns.

References:
[1] Amenta, N.; Kil, Y.-J.: Defining point-set surfaces, ACM Transactions on Graphics, 23(3), 2004, 264-270. https://doi.org/10.1145/1186562.1015713
[2] Dey, T.-K.; Sun J.: An adaptive MLS surface for reconstruction with guarantees, Symposium on Geometry processing, Vienna, Austria, July 4-6, 2005, 43-52.
[3] 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
[4] Lipson, H.; Kurman, M.: Fabricated: The New World of 3D Printing, John Wiley & Sons, Indianapolis, IN, 2013.
[5] 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://dx.doi.org/10.1016/S0010-4485(02)00087-8
[6] 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://dx.doi.org/10.1108/13552540310502185
[7] 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://dx.doi.org/10.1016/S0010-4485(03)00097-6
[8] 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
[9] 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: 10.1115/1.3010710