COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 3, 330–336
https://doi.org/10.1080/16864360.2017.1397884

An improved virtual edge approach to slicing of point cloud for additive manufacturing

Jinting Xu, Wenbin Hou and Hongzhe Zhang

Dalian University of Technology, China

ABSTRACT
At present, the dense and accurate point cloud has been able to be easily acquired via various 3D measurement devices. In this situation, additive manufacturing (AM) based directly on the point cloud has been paid more attention than before. For direct slicing of the point cloud, an improved virtual edge (VE) approach is proposed. In the method, the process of constructing VEs is first discussed briefly, and then a simple algorithm of sorting contour points is given to process the scattered intersections of the VEs and the slicing plane. Subsequently, the additional points are inserted between the sparse contour points and then the contour points are adjusted onto the underlying nominal surface, thereby forming the final layered contours. Since the proposed method does not involve any nonlinear optimization, it is mathematically robust and has the ability of recognizing and processing correctly the multi-contours on a slicing plane. Finally, several examples are given to validate this method.

KEYWORDS
Additive manufacturing; Point cloud; Slicing; Virtual edge; Layered contour

1. Introduction

Additive manufacturing (AM) is an innovative technology to fabricate the parts with complex shapes, lately has been becoming the focus of media and gained the increasing attentions of the researchers in various fields [2]. Unlike the subtractive manufacturing processes, such as milling which removes the unwanted materials from a blank, AM builds up a part through the deposition of the materials layer by layer. This distinctive feature of AM brings many advantages, such as the abilities to manufacture the parts with complex internal structures. It can also fabricate cost-effectively the components made of the expensive materials such as titanium and nickel in aerospace industry, because for such parts, the traditional milling usually suffers from an extremely high material removal ratio [1].

Various AM processes have been developed, including stereolithography apparatus (SLA), fused deposition modeling (FDM), selective laser sintering or melting (SLS/SLM), electron beam melting (EBM), etc. Although the mechanical details of these processes are different from each other, they all require slicing 3D model of the part into a set of 2.5D layered contours along the building direction [12]. So far, the models used widely in AM are the parametric and STL models [8], [11] that both require the model conversion from the point cloud. Although many works have been done for the solution of the parametric or STL model construction, a completely automated and satisfactory solution is still challenging and far from being fully automatic even for the high-quality measured data points [3], [16]. With the development of 3D measuring technologies, the dense and accurate point cloud, which can represent more exactly the geometry of a part than before, has been nicely available. In this situation, the slicing methods have begun to evolve from slicing the parametric and STL models to the methods that slice directly the point cloud as they bypass completely the process of the model conversion from the point cloud [6], [7], [9]. In spite of this, compared with the parametric and STL models, the point cloud only provides the position information of the discrete points of the nominal surfaces so that slicing directly the scattered point cloud is not very easy task.

In the past decade, some valuable works, which devote to slice directly the point cloud, have been proposed, and the methods can be classified mainly onto three categories: projection point band (PPB) based method [4–6], [15], point projection (PP) based method [10], [16], [17] and virtual edge (VE) based method[9], [13]. PPB based method subdivides the point cloud into a number of layers with a thickness perpendicular to the building direction, and then the data points between two neighboring

CONTACT Jinting Xu xujt@dlut.edu.cn; Wenbin Hou houwb@dlut.edu.cn; Hongzhe Zhang zhanghongzhe@dlut.edu.cn

layers are projected onto the layered plane, forming 2D PPB. Subsequently, the data points in PPB are sorted and compressed by keeping the featured points by a digital image based reduction method [5], a reduction method based on oriented vector [5] or a linear correlation based method [15]. Finally, the featured points are linked into a polygon contour or fitted to a B-spline curve [6]. This method is straightforward and easy to be realized, but it can generally result in a trade-off between the projection error and the truncation error [16]. In the PP based method proposed in [10], [16], moving least square (MLS) surface is used to represent the nominal surface of the point cloud and the intersection point of MLS surface and a defined line is computed as the contour point. Since the curvatures can be computed from a set of closed formula based on the MLS surface, the step length of the contour and the thickness of every layer are both adaptively controlled, so the accuracy and efficiency of AM manufacturing are improved, but a relatively complex nonlinear optimization is involved when calculating the layered contours.

Compared with the above methods, VE based method proposed by Sun et al [13] does not involve any complicated calculation, but uses a set of minimum distance-based correlated point pair (MDCPP) to construct VEs and then intersects these VEs with the slicing plane for obtaining the contour points. This method is mathematically straightforward and very easy to be implemented. It is also noted that, in Sun's method [13], one arbitrary point in MDCPP can be not shared by multiple VEs; this may result in the potential accuracy loss of the contour, especially when processing the non-uniformly distributed data points. To complement the weakness of VE approach, Park et al [9] improved the VE based method using the density gauge sphere and the point supplementing technique, but excessive number of VEs may cause vibration of the layered contour so that the generated contour has to be further filtered and smoothed. The key challenge in slicing the point cloud by the VE based approach is to develop robust method to reduce the accuracy loss of the contour caused by the insufficient number of VEs and at the same time address effectively multi-contours on a slicing plane. To do so, an improved VE method is proposed in this paper. The remainder of this paper is organized as follows: Section 2 presents briefly the constructing process of VEs. The processing method of the scattered intersection points of the VEs with the slicing plane is discussed in Section 3. The methods of contour point supplementing and adjusting are proposed in Section 4. Section 5 gives some examples to validate the proposed method. The concluding remarks are presented in Section 6.

2. Initial contour point determination

In this proposed method, the initial contour point are produced by intersecting the VEs with the slicing plane, the scattered intersection points are then sorted and divided into several individual contours if there are multiple contours. Subsequently, the positions of the initial contour points are adjusted onto the underlying nominal surface and additional points are inserted between the sparse contour points, thereby forming the final layered contours.

As shown in Fig. 1, $P^{(i)}$ is the ith slicing plane with height h and the point set between the planes $P_{+}^{(i)}$ $(h+\Delta\delta)$ and the planes $P_{-}^{(i)}(h+\Delta\delta)$ is defined as its correlated point set Cp that consists of two sets of points, i.e., Q_{+} between $P_{+}^{(i)}$ and $P^{(i)}$, and Q_{-} between $P_{-}^{(i)}$ and $P^{(i)}$. $\Delta\delta$ is the width of the correlated domain between the planes $P_{+}^{(i)}$ and $P^{(i)}$ and is usually determined according to the density of the point cloud. For the points $\boldsymbol{q}_i$ in the correlated domain,

$$\begin{cases} \boldsymbol{q}_i \in Q_{+}, \text{if } \boldsymbol{n}_p \cdot (\boldsymbol{q}_i - \boldsymbol{p}_0) > 0 \\ \boldsymbol{q}_i \in Q_{-}, \text{if } \boldsymbol{n}_p \cdot (\boldsymbol{q}_i - \boldsymbol{p}_0) < 0 \end{cases} \tag{1}$$

where $\boldsymbol{p}_0$ is a point lying on the slicing plane $P^{(i)}$ and $\boldsymbol{n}_p$ is the normal vector of $P^{(i)}$. Then, between Q_{+} and Q_{-}, the MDCPPs can be identified by the following procedures:

Step 1. Select one point $\boldsymbol{q}^{(r)}$ from the unvisited points of $Q_{+}(Q_{-})$, and determine its closest point $\boldsymbol{q}^{(t)}$ from $Q_{-}(Q_{+})$, as shown in Fig. 1. If $\boldsymbol{q}^{(t)}$ belongs to the identified already MDCPP, delete $\boldsymbol{q}^{(r)}$ from Cp and go to Step 1; otherwise, go to Step 2.

Step 2. Search the closest point $\boldsymbol{q}^{(s)}$ of $\boldsymbol{q}^{(t)}$ from $Q_{+}(Q_{-})$; If $\boldsymbol{q}^{(s)}$belongs to the identified MDCPP, delete $\boldsymbol{q}^{(r)}$ and $\boldsymbol{q}^{(t)}$ from Cp and go to Step 1; Otherwise, go to Step 3.

Step 3. If $\boldsymbol{q}^{(s)}$ coincides with $\boldsymbol{q}^{(r)}$, $\boldsymbol{q}^{(r)}$ and $\boldsymbol{q}^{(t)}$ is MDCPP; otherwise $\boldsymbol{q}^{(s)}$ and $\boldsymbol{q}^{(t)}$ is MDCPP and delete $\boldsymbol{q}^{(r)}$ from Cp.

Step 4. If no unvisited points in Cp, the algorithm terminates; otherwise go to Step 1.

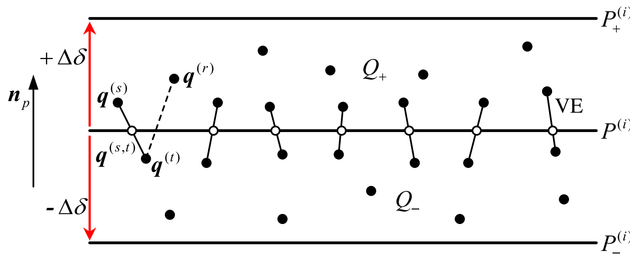

Figure 1. Initial contour point determination based on VE.

Assume that $(\boldsymbol{q}^{(s)}, \boldsymbol{q}^{(t)})$ is one MDCPP, the intersection, i.e. initial contour point, of the connecting line of $\boldsymbol{q}^{(s)}$ and $\boldsymbol{q}^{(t)}$ with the slicing plane $P^{(i)}$ can be calculated by the following equation,

$$\boldsymbol{q}^{(s,t)} = \boldsymbol{q}^{(t)} + \frac{(\boldsymbol{p}_0 - \boldsymbol{q}^{(t)}) \cdot \boldsymbol{n}_p}{(\boldsymbol{q}^{(s)} - \boldsymbol{q}^{(t)}) \cdot \boldsymbol{n}_p} (\boldsymbol{q}^{(s)} - \boldsymbol{q}^{(t)}) \tag{2}$$

3. Initial contour construction

VEs are randomly constructed in the above procedures so that the initial contour points are usually scattered and non-uniformly distributed, which need to be further sorted and be connected into a closed contour. If there are two or more contours on a slicing plane, each individual contour also needs to be recognized and divided correctly. In this paper, a tracing method based on angle detection is proposed to sort the initial contour points and recognize all individual contours.

Assume that the initial contour points set is P_{C}. The point in $\mathrm{P_C}$, which has the minimum x coordinate, is selected as the starting point of the contour to be searched and denoted by $\boldsymbol{p}_{\mathrm{C},0}$. A moving frame, on the slicing plane, is mounted on $\boldsymbol{p}_{\mathrm{C},0}$, as shown in Fig. 2, and this frame is represented by

$$\{\boldsymbol{p}_{\mathrm{C},0}; \boldsymbol{e}_1^{(\mathrm{M})}, \boldsymbol{e}_2^{(\mathrm{M})}\} \tag{3}$$

where the coordinate axes $\boldsymbol{e}_1^{(\mathrm{M})}$ and $\boldsymbol{e}_2^{(\mathrm{M})}$ are initially consistent respectively with X and Y axes of the model coordinate system. Then, the next contour point is identified if the angle φ_i is minimum between the negative axis of $\boldsymbol{e}_1^{(\mathrm{M})}$ and the line that connects $\boldsymbol{p}_i$ and $\boldsymbol{p}_{\mathrm{C},0}$, where $\boldsymbol{p}_i$ is one point in K-neighbor of $\boldsymbol{p}_{\mathrm{C},0}$ and φ_i is calculated by the following equation,

$$\varphi_i = \begin{cases} \cos^{-1}\left(-\dfrac{\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{C},j-1}}{\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{C},j-1}} \cdot \boldsymbol{e}_1^{(\mathrm{M})}\right), & \boldsymbol{p}_i \in \mathrm{I, II} \\ \quad \times \text{quadrants} & \\ 2\pi - \cos^{-1}\left(-\dfrac{\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{C},j-1}}{\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{C},j-1}} \cdot \boldsymbol{e}_1^{(\mathrm{M})}\right), & \boldsymbol{p}_i \in \mathrm{III, IV} \\ \quad \times \text{quadrants} & \end{cases} \tag{4}$$

where $\boldsymbol{p}_{\mathrm{C},j-1}$ stands for the extracted jth contour point. Once the contour point $\boldsymbol{p}_{\mathrm{C},j}$ is gained, it is used as the new starting point for finding the next contour point. The direction vector of the line form $\boldsymbol{p}_{\mathrm{C},j-1}$ to $\boldsymbol{p}_{\mathrm{C},j}$ is used as $\boldsymbol{e}_1^{(\mathrm{M})}$axis of the jth moving frame mounted on $\boldsymbol{p}_{\mathrm{C},j}$. Then, the procedures of searching contour points are performed repeatedly until the starting point $\boldsymbol{p}_{\mathrm{C},0}$ is reached again. Here, it is assumed that the noise of the point cloud has been removed, although filtering noise is a very critical issue, it beyond the scope of this paper. When processing such point cloud, the initial contour points are almost in a underlying curve. In this situation, the starting point $\boldsymbol{p}_{\mathrm{C},0}$, namely the one having the minimum x coordinate, satisfies Eq. (4), this ensures that $\boldsymbol{p}_{\mathrm{C},0}$ can be reach again. After a closed contour is gained by the above algorithm, the extracted contour points, as well as the points that are inside this contour, are removed from the initial contour point set $\mathrm{P_C}$. After that, if $\mathrm{P_C}$ is not empty, the remaining points in $\mathrm{P_C}$ will be used as a new point set $\mathrm{P_{C,new}}$ to initiate the above sorting procedure to extract the next individual contour. The starting point of the next contour is still selected the one which has the minimum x coordinate in $\mathrm{P_{C,new}}$. The above sorting and recognizing procedures are repeated iteratively until all points in $\mathrm{P_C}$ are all processed. As a result, all individual contours can be automatically recognized.

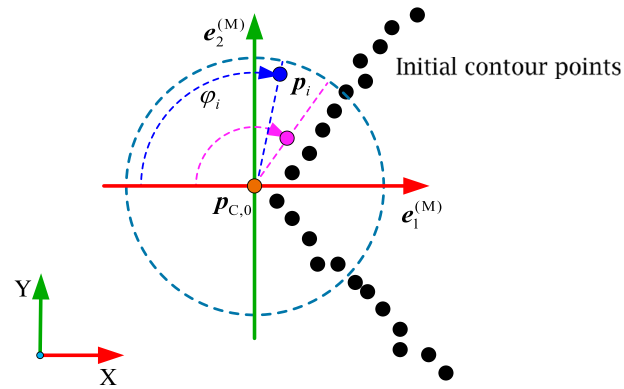

Figure 2. The initial contour extraction.

It is noted that, in the above procedures, the selection of K-neighbor of the current contour point $\boldsymbol{p}_{\mathrm{C},j}$ plays a vital role for identifying the next contour point. Generally, selection of K value depends on the resolution of the initial contour points. If the common industry scanning devices, such as ATOS-II in our lab, are used, K is in the range of 10-15, because the initial contour points are almost in an underlying curve, compared with the scattered measured points. In our experiments, such K can generally produce a satisfactory result. In addition, the neighboring points of $\boldsymbol{p}_{\mathrm{C},j}$, have also important influence on recognition of the multiple contours. In order to exclude as much as possible the points that are in K-neighbor points above but are the points of other contours, a threshold of the distance, d_{c}, between two neighboring contours needs to be specified in advance. The basic requirement is that d_{c} is not less than the density of the point cloud. In our experiments, d_{c} is set be

$$d_{\mathrm{c}} = (10 - 15)\rho \tag{5}$$

where ρ is the density of the point cloud. Constrained by d_c, K-neighbor of $\boldsymbol{p}_{C,j}$ can be rewritten as

$$K(\boldsymbol{p}_{C,j}) = \{\boldsymbol{p}_i; \boldsymbol{p}_i - \boldsymbol{p}_{C,j} \le d_c, i = 1, \cdots, m_K, m_K \le K\} \tag{6}$$

Generally, using Eq. (6) can avoid adding the points, which is other contours and the distance from $\boldsymbol{p}_{C,j}$ is more than d_c, into K-neighbor of $\boldsymbol{p}_{C,j}$, thus guaranteeing each individual contour on the same slicing plane can be recognized.

4. Contour point supplement and adjusting

It is noted that, limited by the size of K-neighbor points and the density distribution of the point cloud, a large 'gap' between two consecutive contour points may appears, like the dotted line shown in Fig. 3. If the gap is too large, it can not be ignored in generating the layered contour because it might possibly lead to the loss of the detail feature. Besides, the initial contour points are only the intersections of the VEs and the slicing plane, so it is difficult to guarantee that the intersections can lie accurately on the underlying nominal surface. Generally, the lower the density of the point cloud, the lager the tolerance between the intersections and the nominal surface. In this situation, the position of the initial contour point needs to be adjusted so that it can lies on the nominal surface as accurately as possible, and the large gap between two consecutive contour points also requires inserting additional contour points for generating the desired layered contours.

4.1. Additional point supplementing

The gap, which needs to insert additive contour points, is first identified if its width exceeds the threshold of the distance d_g. d_g is determined by the following equation,

$$\begin{cases} d_g = \min\{d_s, d_m\} \\ d_m = \left(\left(\sum_{i=1}^{n-1} \|\boldsymbol{p}_i - \boldsymbol{p}_{i+1}\| \right) + \|\boldsymbol{p}_n - \boldsymbol{p}_1\| \right) \Big/ n \end{cases} \tag{7}$$

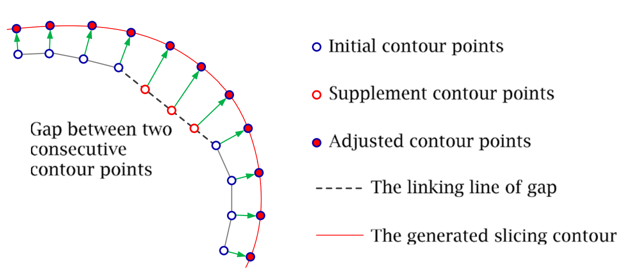

Figure 3. Initial contour point adjusting and supplement.

where d_s is the allowable minimum width of the gap specified by the user, d_m is the mean length of all segments of the contour and $\boldsymbol{p}_i$ denotes the initial contour point. For the identified gap, a line segment is added to link the two end points of the gap, as shown in Fig. 3, and then some additional points are inserted on this line segment. The number of the additional points to be inserted, m, is calculated by

$$m = \left[0.99 \times \frac{w_{gap}}{d_g} \right] \tag{8}$$

where $[\cdot]$ denotes the integer, w_{gap} is the width of the gap and the coefficient, 0.99, ensures that the gap, whose width is more than d_g a little, does not insert points. The points $\boldsymbol{p}_{insert,j}$, to be inserted, is determined by

$$\boldsymbol{p}_{insert,j} = \boldsymbol{p}_s + \frac{j}{m}(\boldsymbol{p}_e - \boldsymbol{p}_s) \tag{9}$$

In Eq. (9), $\boldsymbol{p}_s$ and $\boldsymbol{p}_e$ are the start and end points of the gap, respectively. After inserting the additional points a relatively dense and uniformly distributed contour points sequence is obtained.

4.2. Position adjusting of contour point

In the section, the contour points will be adjusted as accurately as possible onto the underlying surface. Assume that $\boldsymbol{q}_j, j = 0, 1, \cdots, k$ is the K-neighbor points of the contour point $\boldsymbol{p}_i$. A simple quadric surface is used to approximate the local nominal surface of K-neighbor points. First, a local coordinate system,

$$\xi^{(L)} = \{\boldsymbol{p}_i; \boldsymbol{e}_1^{(L)}, \boldsymbol{e}_2^{(L)}, \boldsymbol{e}_3^{(L)}\} \tag{10}$$

is mounted on $\boldsymbol{p}_i$ where $\boldsymbol{e}_3^{(L)}$ is the unit normal vector of the surface at $\boldsymbol{p}_i$ which can easily obtained by using eigenvalue analysis of the covariance matrix of the neighbor points, $\boldsymbol{e}_1^{(L)}$ is the unit vector of the projection of the line linking $\boldsymbol{p}_i$ and arbitrary one point in its K-neighbor on the tangent plane and $\boldsymbol{e}_2^{(L)}$ is the cross product of $\boldsymbol{e}_3^{(L)}$ and $\boldsymbol{e}_1^{(L)}$. Then, K-neighbor points $\boldsymbol{q}_j$ is transformed into $\boldsymbol{q}_j^{(L)}$ in $\xi^{(L)}$.

Assume that the quadric surface, to be fitted, is $z^{(L)} = z(x^{(L)}, y^{(L)})$, then the least squares objective function of fitting the quadric surface to the K-neighbor points can be written as

$$E = \min \left[\sum_{j=1}^{k} w_j \left\| z^{(L)}(x_j^{(L)}, y_j^{(L)}) - z_j^{(L)} \right\|^2 \right] \tag{11}$$

where w_j is weight of $\boldsymbol{q}_j^{(L)}$. Generally, the closer $\boldsymbol{q}_j^{(L)}$ from $\boldsymbol{p}_i$, the larger its weight, thus the following weight

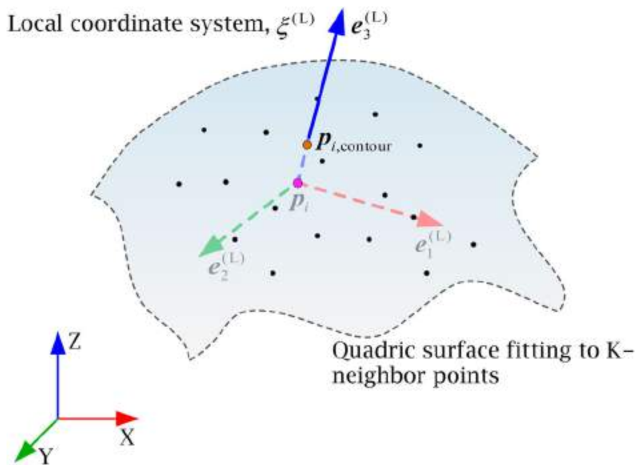

Figure 4. Position adjusting of the contour point.

function is adopted

$$w_j = \exp\left[-d_j/\left(\frac{1}{k}\sum_{r=1}^{k} d_r\right)\right] \tag{12}$$

In Eq. (12), d denotes the distance from one point in K-neighbor point set to the contour point $\boldsymbol{p}_i$. Once the quadric surface is fitted, the initial contour point can be adjusted onto the nominal surface, as shown in Fig. 4, using the following equation

$$\boldsymbol{p}_{i,\mathrm{contour}} = \boldsymbol{p}_i + z^{(L)}(0,0)\boldsymbol{e}_3^{(L)} \tag{13}$$

5. Experiments

The algorithms proposed in this paper have been coded in C++ language and implemented in a PC. In the following, two examples are given to demonstrate respectively the accuracy of the proposed method and its ability of processing the complex model.

5.1. Test for the accuracy of the proposed method.

In this section, the accuracy of the generated contour point by the proposed method is tested. A B-spline surface, $\boldsymbol{S}(u,v)$, as shown in Fig. 5, is constructed and then 10, 000 points are sampled from this surface. Three parallel planes are selected to slice the point cloud. The VE method in [13] and the improved VE method are applied respectively to calculate the contour points. The theoretical contour point is calculated by projecting the initial contour point onto the surface by solving Eq. (14) using constrained optimization methods [14].

$$\begin{cases} \min\limits_{u,v} d(u,v) \\ s.t.(\boldsymbol{S}(u,v) - P_0)\cdot \boldsymbol{n}_P = 0 \end{cases} \tag{14}$$

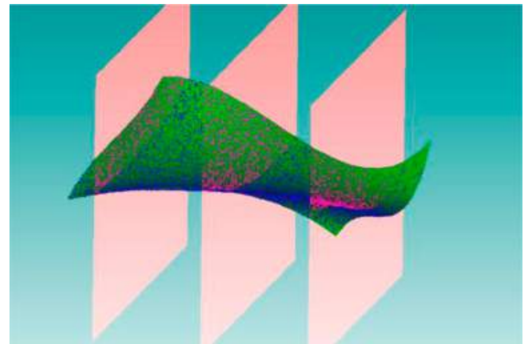

Figure 5. Test for the accuracy of the generated contour by the proposed method.

Table 1. Comparison of the improved VE method and the VE method in [13]

Methods	Minimum error (mm)	Maximum error (mm)	Average error (mm)
Improved VE method	0.011	0.153	0.054
VE method in [13]	0.038	0.628	0.150

where $d(u,v) = (\boldsymbol{S}(u,v) - \boldsymbol{p}_i)\cdot(\boldsymbol{S}(u,v) - \boldsymbol{p}_i)$ is the squared distance function from a initial contour point $\boldsymbol{p}_i$ to $\boldsymbol{S}(u,v)$, P_0 is a point lying on the slicing plane and $\boldsymbol{n}_P$ is the normal vector of the slicing plane. The accuracy the method is the average error of all contour points which is defined by

$$\epsilon_{\mathrm{cpoint}} = \sum_{j=1}^{n_c} \|\boldsymbol{q}_j - \boldsymbol{q}_j^*\| \Big/ n_c \tag{15}$$

where $\boldsymbol{q}_j$ and $\boldsymbol{q}_j^*$ are the calculated contour point and the corresponding theoretical contour point.

The comparison of the VE method in [13] and the improved VE method of this paper is performed and the results are listed in Table 1. From Table 1, it is seen that, the minimum, maximum and average errors of the VE method in [13] are 0.038 mm, 0.628 mm, 0.150 mm, respectively, and the errors of the improved VE method by us are 0.011 mm, 0.153 mm, 0.054 mm. Regardless of the average, maximum or minimum errors, our method all hold the good performance, and an improvement of average error of about 64% is obtained. This shows that the improved VE method by us can improve the accuracy of the generated layered contours.

5.2. Slicing the complex point cloud

In the following, four examples are presented to demonstrate the ability of the proposed method to slice the complex cloud of point. The point clouds for test are shown in Fig. 6. To display clearly the generated layered contours, a big slicing thickness is used in the

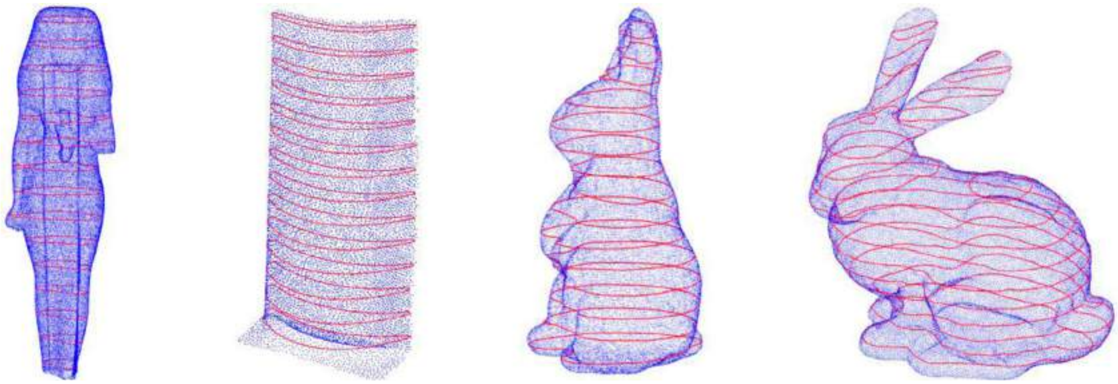

Figure 6. Examples of slicing the complex point cloud using the proposed method.

experiments. From the slicing results, it is seen that our method can generate nicely the layered contours, not requiring any model conversions from the cloud of points and the problem of the multi-contours when processing the complex point clouds can be also recognized automatically and divided correctly, which has been demonstrated by the experimental results shown in Fig. 6.

6. Conclusions

An improved VE method is proposed in this paper. In the proposed method, the gap problem between two consecutive contour points, which results from the non-uniform density distribution of the point cloud, is solved by supplying additional points, and the accuracy of the contour points is improved by the developed position adjusting strategy. This has been demonstrated by our experiments in which, compared with the traditional VE method, the accuracy of the generated contour points by our method is improved about 64%. In addition, the sorting algorithm for the scattered contour points not only guarantees that the scattered contour point can be connected into a closed contour, but also makes the proposed method have the ability of recognizing and extracting correctly the multiple contours on a same slicing plane. This point is also confirmed by the given examples.

Acknowledgement

This work was supported by the NSFC (Grant No. 51575086).

ORCID

Jinting Xu http://orcid.org/0000-0002-5280-0929
Wenbin Hou http://orcid.org/0000-0002-3200-1724
Hongzhe Zhang http://orcid.org/0000-0003-4910-0689

References

[1] Ding, D.; Pan, Z.; Cuiuri, D.; Li, H.; Larkin, N.; Duin, S.: Automatic multi-direction slicing algorithms for wire based additive manufacturing, Robotics and Computer-Integrated Manufacturing, 37, 2016, 139–150. http://doi.org/10.1016/j.rcim.2015.09.002

[2] Gao, W.; Zhang, Y.; Ramanujan D.; Ramania, K.; Chen, Y.; Williams, C. B.; Wang, C.; Shin, Y.; Zhang, S.; Zavattieri P. D.: The status, challenges, and future of additive manufacturing in engineering, Computer-Aided Design, 69, 2015, 65–89. http://doi.org/10.1016/j.cad.2015.04.001

[3] Li, Y.; Ni, J.: Constraints based nonrigid registration for 2D blade profile reconstruction in reverse engineering, ASME, Journal of Computing and Information Science in Engineering, 9(3), 2009, 031005-031005-9. http://doi.org/10.1115/1.3184602

[4] Liu, G. H.; Wong, Y. S.; Zhang, Y. F.; Loh, T. H.: Modelling cloud data for prototype manufacturing, Journal of Materials Processing Technology, 138(1), 2003, 53–57. http://doi.org/10.1016/S092 4-0136(03)00048-7

[5] Liu, G. H.; Wong, Y. S.; Zhang, Y. F.; Loh, T. H.: Error-based segmentation of cloud data for direct rapid prototyping, Computer-Aided Design, 35(7), 2003, 633–645. http://doi.org/10.1016/S00 10-4485(02)00087-8

[6] Javidrad, F.; Pourmoayed, A. R.: Contour curve reconstruction from cloud data for rapid prototyping, Robotics and Computer-Integrated Manufacturing, 27(2), 2011, 397–404. http://doi.org/10.1016/j.rcim.2010.08.008

[7] Kumbhar, V. K.; Pandey, P. M.; Rao, P. V. M.: Improved intermediate point curve model for integrating reverse engineering and rapid prototyping, International Journal of Advanced Manufacturing Technology, 37(5-6), 2008, 553–562. http://doi.org/10.1007/s00170-007-0995-y

[8] Pandey, P. M.; Reddy, N. V.; Dhande, S. G.: Real time adaptive slicing for fused deposition modeling, International Journal of Machine Tools & Manufacture, 43(1), 2003, 61–71. http://doi.org/10.1016/S0890-6955(02)00164-5

[9] Park, H. T.; Chang, M. H.; Park, S. C.: A slicing algorithm of point cloud for rapid prototyping, Proceedings of the 2007 Summer Computer Simulation Conference, San Diego, California, USA, July, 2007, pp. 24.

[10] Qiu, Y.; Zhou, X.; Qian, X.: Direct slicing of cloud data with guaranteed topology for rapid prototyping,

International Journal of Advanced Manufacturing Technology, 53(1-4), 2011, 255–265. http://doi.org/10.1007/s00170-010-2829-6

[11] Starly, B.; Lau, A.; Sun, W.; Lau, W.; Bradbury, T.: Direct slicing of STEP based NURBS models for layered manufacturing, Computer-Aided Design, 37(4), 2005, 387–397. http://doi.org/10.1016/j.cad.2004.06.014

[12] Steuben, J. C.; Iliopoulos, A. P.; Michopoulos, J. G.: Implicit slicing for functionally tailored additive manufacturing, Computer-Aided Design, 77, 2016, 107–119. http://doi.org/10.1016/j.cad.2016.04.003

[13] Sun, Y. W.; Guo, D. M.; Jia, Z. Y.; Liu W. J.: B-spline surface reconstruction and direct slicing from point clouds, International Journal of Advanced Manufacturing Technology, 27(9-10), 2006, 918–924. http://doi.org/10.1007/s00170-004-2281-6

[14] Tang, H. W., Qin, X. Z.: Practical methods of optimization. Dalian: Dalian University of Technology Press; 2004. (In Chinese)

[15] Wu, Y. F.; Wong, Y. S.; Loh, H. T.; Zhang, Y. F.: Modelling 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

[16] Yang, P.; Qian, X.: Adaptive slicing of moving least squares surfaces: toward direct manufacturing of point set surfaces, ASME, Journal of Computing and Information Science in Engineering, 8(3), 2009, 031003-031003-11. http://doi.org/10.1115/1.2955481

[17] Zhang, Y.; Ge, L.: Adaptive tool-path generation on point-sampled surfaces. Precision Engineering, 35(4), 2011, 591–601. http://doi.org/10.1016/j.precisioneng.2011.04.002