An adaptive slicing approach for processing STL massive data model in batches based on layer merging

Minghao Shao¹², Chao Wei¹², Bin Cui¹², Yongkang Li¹² and Tengfei Zheng¹²

¹ School of Mechanical Engineering, Xi’an Jiaotong University, Xi’an 710049, China
² State Key Laboratory for Manufacturing and Systems Engineering, Xi’an 710049, China
*Corresponding author e-mail: cuibin@xjtu.edu.cn

Abstract. Slicing is one of the core parts of the additive manufacturing software system, which completes the function of transforming the 3D model into a 2D profile. The adaptive slicing algorithm uses different layer thicknesses for layering in different areas according to the changes in the geometry of the mesh model. Where the model is more complex and the curvature changes a lot, a smaller layer thickness is used to approximate the model. In areas of large curvature of the model, maximum layer thickness is used to improve printing efficiency. With the development of additive manufacturing technology, grid model files such as STL have become larger and larger, and the amount of data needed to be processed by slicing software has increased dramatically. Existing slicing software is limited by computer hardware and cannot process massive data model. It greatly limits the development of additive manufacturing technology, so it is urgent to propose a new method to solve the rapid stratification of mass data model. In this paper, we proposed an adaptive slicing approach for processing STL massive data model in batches based on layer merging. At the same time, it is compared with the fixed-layer thickness slicing method, which shows that the algorithm can improve printing efficiency under the premise of ensuring accuracy.

1. Introduction

In recent years, 3D printing technology has developed rapidly. The manufacturing industry not only puts forward higher requirements for the accuracy of additive manufacturing, but also pursues higher manufacturing efficiency. Slicing is one important step of 3D printing. It is closely related to the build time and surface quality of the parts [1]. The task of slicing is to translate 3D models into series of 2D contours. The solid model is represented by series of layers. Each layer is made up of interior and exterior boundary polylines. These data are called as sliced data of the model. Stereolithography (STL) mesh model file is an intermediate form of 3D model and has become the standard in additive manufacturing [2, 3]. Most current commercial additive manufacturing systems use this format file for slicing and manufacturing [4, 5].

Although there are many factors that affect the printing time, no matter how the layer thickness and cross-sectional area change, the printing time of each layer is almost constant. And the time related to the hardware cannot be optimized, so the only factor that can be optimized that affects the printing time is the number of layers. A simple way to change the number of layers is to increase the thickness of the layers. Although this method reduces the number of printed layers of the input model, the discontinuous boundaries between layers greatly reduce the print quality. The optimized slicing can reduce the manufacturing time of the parts and improve the surface quality of the parts [6], and the adaptive slicing
algorithm has emerged. According to the geometric characteristics of the STL model, Li H [7] et al. used a thin thickness for areas with large curvature changes, and thick thicknesses for areas with small curvature changes. By calculating the normal vector of the triangular patch and the Z axis to determine the delamination height, the acceptable tip error needs to be specified in advance. Xiaodi Pan [8] et al. studied the adaptive slicing of the STL model. Unlike other adaptive slicing methods, this paper uses the area ratio between two adjacent layers to determine the layer thickness. If the ratio is small, it can be considered there is little change between the two layers, and a thick layer thickness can be adopted. It is necessary to set an allowable area deviation ratio in advance. If the ratio between the areas of the two layers is large, the current layer thickness needs to be reduced. Li Q [9] et al. proposed an adaptive slicing method for functionally graded material models. The material information is also taken into account in the slicing process. When calculating the layer thickness, not only the tip height error of the model geometry information, but also the material gradient is considered. Finally, these two parameters were weighed to calculate the final stratification thickness.

With the development of additive manufacturing technology, mesh model files such as STL have become larger and larger, and the amount of data needed to be processed by slicing software has increased dramatically. Existing slicing software is limited by computer hardware and cannot process massive data models (>5GB). In order to perform adaptive slicing of massive mesh models, an adaptive slicing method based on layer merging is proposed to process massive mesh model in batches. First, we divide the model along the printing direction to obtain a series of small data volume sub-models. The data volume of the sub-models needs to be less than the memory of the computer to read the data at a time. Then slice the sub-model with the minimum layer thickness, and analyze the similarity of the slice information of two adjacent layers with the method in article [6]. Combine two adjacent slices until the similarity is greater than the set value, which realizes adaptive slice based on layer merging. Finally, the sliced sub-models are merged to obtain the adaptive layering information of the massive data model. At the same time, we printed a case model to verify the effectiveness of this method.

2. Mechanism of the adaptive slicing approach for processing STL massive data model in batches based on layer merging

2.1 Segmentation of the STL mesh model

When processing STL model files, the computational memory will swell about 5 times. For example, to read a 1GB STL model file, about 5GB of computer memory is needed to successfully complete the hierarchical processing. But for the sake of safety, we set safety factor is 1.2, that means 6 times the memory space of the read file size is used to process the model. So, when the available memory is A, the maximum file size (Ms) that this algorithm can handle is A/6. For the STL model file to be divided, group all the triangular patches according to the size of the Z coordinate. First traverse the STL model file to find the model's maximum Z-direction coordinate Zmax and minimum Z-direction coordinate Zmin, and divide them into intervals. The number of intervals is equal to the size of the STL model file divided by Ms, and each triangle patch is based on its own Z coordinate. Place them in the corresponding interval to form multiple sub-models. If there is a sub-model file size that is still larger than Ms, continue to divide until its size is smaller than Ms.

To study the effectiveness of this segmentation method, we segmented a binary STL model with a file size of 7.05G as shown in Figure 1. The model contains 151,616,938 triangular patches with a size of 449mm×336mm×428mm, and the maximum readable STL file size is 1.2GB. Firstly, the model is divided into seven sub-models of about 1GB, and then the seven sub-models are layered to obtain seven temporary layered results. The temporary layered results are stored in the hard disk file to reduce the memory usage, and finally the seven temporary layering result files are combined to get the final layering result of this massive data model.
2.2 The method of layer merging

This section presents the adaptive slicing approach of layer merging. First, the sub-models is sliced to the smallest layer thickness, and after layering, the polygon outline of each layer is retained. Then determine whether to merge the two layers by comparing the similarity of the polygons between the two adjacent layers. For any two adjacent layers, calculate the Ameliorative Area Ratio (AAR) and Variation of the Cusp Height (VCH) [6] by the Boolean operation of the polygon. When any one of the two parameters is greater than the corresponding allowable value, this one layer should be retained to ensure surface quality. When both parameters are less than the allowable value, it indicates that the surface accuracy is higher than the requirement at this time. This layer and the previous layer should be combined to form a layer to improve manufacturing efficiency.

The first step is to compare AAR with the maximum allowable value. If the AAR is greater than the allowable value, the layer is retained, otherwise the second step is performed. The second step is to calculate VCH for further evaluation. If the VCH is greater than the allowable value, the layer should be retained, otherwise the layer should be removed. Since the layer thickness cannot be greater than the maximum allowable layer thickness, the layer thickness removed in the second step is judged in the third step. If the layer thickness is greater than the maximum allowable layer thickness, keep the layer. Once the layer is confirmed as reserved, the comparison ends, and the next layer is compared until the top layer of the model is reached.

For the adjacent \(i\) layer and \((i+1)\) layer, the \(i\) layer is selected as the target area, and the \((i+1)\) layer is used as the clipping area of the polygonal Boolean operation. Then AAR is expressed as:

\[
\mathcal{E} = \begin{cases} 
\frac{S_{\text{sub}} + S_{\text{clip}} - 2S_{\text{ins}}}{S_{\text{clip}}} & S_{\text{sub}} \geq S_{\text{clip}}, S_{\text{clip}} \neq 0 \\
\frac{S_{\text{sub}} + S_{\text{clip}} - 2S_{\text{ins}}}{S_{\text{sub}}} & S_{\text{sub}} < S_{\text{clip}}, S_{\text{sub}} \neq 0 \\
\inf & S_{\text{sub}} = 0 \text{ or } S_{\text{clip}} = 0 
\end{cases}
\]  

(1)
Where: $S_{sub}$ — the area of the $i$ layer; $S_{clip}$ — the area of the $(i+1)$ layer; $S_{ins}$ — the area after the Boolean intersection of two adjacent layers of polygons.

The actual area $S_{lay}$ of each layer is equal to the sum of the area of the polygons in the layer, the area of the filled area is positive, and the area of the hole area is negative. The calculation method of $S_{lay}$ is:

$$S_{lay} = \sum_{i=1}^{k} S_{pi}$$  \hspace{1cm} (2)

Where: $k$ — the number in this layer; the total area of polygons in this layer; $S_{pi}$ — the area of the $i$ polygon. The $S_{pi}$ is:

$$S_p = \frac{1}{2} \left[ (x_1y_2-x_2y_1) + (x_2y_3-x_3y_2) + \cdots + (x_{n-1}y_n-x_ny_{n-1}) + (x_ny_1-x_1y_n) \right]$$ \hspace{1cm} (3)

Where: $x_i$ and $y_i$ are the coordinates of the vertices of the polygon; The area $S_p$ reflects the actual area of the polygon, the inner boundary area is negative and the outer boundary area is positive.

Another parameter is VCH, which is used to represent the residual height of the layered data. VCH can be expressed as the average distance $\bar{r}$ between the outer boundaries of adjacent layers:

$$\bar{r} = \left\{ \begin{array}{ll} \frac{S_{sub} + S_{clip} - 2S_{ins}}{l_{outer}} & l_{outer} \neq 0 \\ \inf & l_{outer} = 0 \end{array} \right. \hspace{1cm} (4)$$

Where: $l_{outer}$ — the perimeter of the outer boundary of the intersecting polygon/ mm; $l_{outer}=0$ means that there are no intersecting polygons. In this case, the residual height is very large, so the average distance is set to infinity.

The maximum allowable VCH can be calculated by the maximum allowable residual height $\delta_0$:

$$r_0 = h_{min} \frac{\delta_0}{\sqrt{h_{min}^2 - \delta_0^2}}$$ \hspace{1cm} (5)

Where: $\delta_0$ — the maximum allowable residual height of the solid model/ mm; $h_{min}$ — the minimum layer thickness/ mm; $r_0$ — the maximum allowable VCH/ mm. In the original layered data, the layered thickness is $h_{min}$, but because some layers are deleted, the layered thickness will change, and the allowable distance of a layer is related to the layered thickness. The maximum allowable distance of a certain layer is calculated as:

$$r_{0i} = r_0 \frac{h_{min}}{h_{i-1}}$$ \hspace{1cm} (6)

Where: $r_{0i}$ — the maximum allowable VCH/mm of the $i$ layer; $h_{i-1}$ — the layer thickness of the $i-1$ layer/mm. When $\bar{r}<r_{0i}$, the $i$ layer can be removed to increase the layer thickness. When $\bar{r} \geq r_{0i}$, the $i$ layer should be reserved.

3. Case study and experiments

3.1. Implementation and results

Table 1 is the hardware platform configuration for algorithm implementation. The operating system used is windows 10, the programming software is QT5.11.1, and the programming language used is C++. As shown in Figure 2, taking the hemispherical model as an example, suppose the minimum layer thickness is 0.2mm, the maximum layer thickness is 0.4mm, the maximum allowable residual height is set to 0.07mm, and the maximum allowable AAR is 0.3. The adaptive slicing result of the model as shown in Figure 3. The upper part of the model is a hemisphere, and the step effect becomes more obvious as the upper end, so the layer thickness is gradually reduced until the minimum layer thickness...
is used for printing to improve the printing accuracy. The results show that the effect of the algorithm is very consistent with the adaptive slicing logic.

Table 1. Hardware system configuration

| Model       | Frequency | Memory | Core number |
|-------------|-----------|--------|-------------|
| CPU         | Intel i9-9900k | 3.60GHz | 8           |
| RAM         | GALAX DDR4  | 2400MHz | 16GB        |
| Graphics card | Titan V      | 1200MHz | 12GB 5120  |

Figure 2. The hemispherical model (the radius is 20mm)

(a) Slicing result with minimum thickness         (b) Slicing result with adaptive thickness

Figure 3. The slicing result of hemisphere model

3.2. Experiment
This experiment is to study the influence of the adaptive layering proposed in this paper on the printing accuracy and printing time of parts. The case model to be printed is shown in Figure 2, with a radius of 20mm. The experiment is divided into four groups in total, using the slicing program written in this article to slice the model with 0.4mm, 0.3mm, and 0.2mm and adaptive thickness. In the adaptive slicing algorithm, the maximum layer thickness is set to 0.4mm, and the minimum layer thickness is 0.2mm. The stratification result is shown in Figure 4.
The printing equipment used in the experiment is Zhuhai Tianwei's desktop FDM printer (CoLiDo D1315 Plus), the material used is PolyLactic Acid (PLA), and the polylactic acid wire diameter is 1.75mm. In the experiment, this algorithm is used to layer the model to generate SLC files, and then the self-edited software of this research group is used to convert the SLC data into machine-readable G codes. The printing parameters used in the experiment are shown in Table 2. Due to its variable layer thickness, adaptive layering needs to adjust the material feed rate of each layer according to its layer thickness. Use the REPETIER-HOST software matched with the printer to load the G code for printing. The time required for the printing process can be directly read by the REPETIER-HOST software.

Table 2. Printing parameters

| Fill rate | Wall thickness /mm | Filling speed /mm·s⁻¹ | Nozzle temperature /°C |
|-----------|--------------------|------------------------|------------------------|
| 20%       | 1.2                | 50                     | 190~210                |

The printed result is shown in Figure 5.

After printing, use a three-coordinate measuring machine (model is Global classic Sr0575) to measure its surface profile. Using the center of the hemisphere as the coordinate origin, select two perpendicular generatrixes on the surface of the hemisphere model, and use a three-coordinate measuring instrument to measure the coordinate values of different points of each curve (as shown in Figure 5(b)). The number of scattered points of the measurement is 280.
In order to measure the surface quality of the printed parts, the error between the sampling point and the theoretical point is calculated, and the variance value $\varepsilon$ of the sampling data is obtained as a measure of the surface quality. The calculation formula of $\varepsilon$ is:

$$
\varepsilon = \frac{1}{N} \sum_{i=1}^{N} \left( \sqrt{x_i^2 + z_i^2} - R \right)^2
$$

(7)

Where: $N$ — the number of scattered points; $i = 1, 2, 3, \ldots, N$; $x_i$ and $z_i$ — scattered points in the first quadrant projection coordinates; $R$ — the theoretical radius of the hemispherical model.

The $R$ in formula (7) is the original radius set when modeling in the 3D software, not the radius of the STL model before the model is printed. The radius of the original model and the radius of the STL model is different, and there is a discrete error between them. Therefore, the calculation of $\varepsilon$ actually includes the discrete error in it. The calculation result of $\varepsilon$ is shown in Table 3.

| Layer thickness /mm | Number of layers/layer | Residual height /μm | Printing time /min | Measured error /μm |
|---------------------|------------------------|---------------------|-------------------|-------------------|
|                     |                        | Maximum             | Average           |                   |
| 0.2                 | 100                    | 50                  | 1.3               | 62.5              | 183               |
| 0.3                 | 67                     | 110                 | 3.5               | 42.3              | 197               |
| 0.4                 | 51                     | 120                 | 5.2               | 31.8              | 247               |
| 0.2-0.4(adaptive)   | 58                     | 50                  | 2.9               | 40.1              | 187               |

It can be seen from the results that the printing time of adaptive layering is between the printing time of 0.2mm layering and 0.4mm layering, and its accuracy is also between the two. The printing time and error of adaptive layering are also smaller than that of using 0.3mm layering. In other words, adaptive layering can take into account both printing time and printing accuracy. It is neither like using the smallest layer thickness but only focusing on the printing accuracy and ignoring the printing efficiency, nor is it like using the maximum layer thickness only focusing on the printing efficiency and ignoring the Printing accuracy.

4. Conclusions

The adaptive slicing algorithm uses different layer thicknesses for layering in different areas according to the changes in the geometry of the mesh model. With the development of additive manufacturing technology, grid model files such as STL have become larger and larger, and the amount of data needed
to be processed by slicing software has increased dramatically. Existing slicing software is limited by computer hardware and cannot process massive data model. In order to perform adaptive slicing of massive mesh model, we proposed an adaptive slicing approach for processing STL massive data model in batches based on layer merging. At the same time, we printed a case model to verify the effectiveness of this method. The results show the method we proposed can not only adaptively layer the STL massive data mesh model, but also improve the printing efficiency under the premise of ensuring the accuracy.

Acknowledgments
The authors are grateful for the support provided by National key research and development program of China (Grant No: 2018YFB1105302), National Natural Science Foundation of China (No.51805427) and the National Science Foundation for Post-doctoral Scientists of China (No.2019M653594).

References
[1] Hu, J. (2017) Study on stl-based slicing process for 3d printing. In: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium. Austin TX. 7-9.
[2] Zha, W., & Anand, S. (2015) Geometric approaches to input file modification for part quality improvement in additive manufacturing. Journal of Manufacturing Processes, 20(OCT.PT.3):465-477.
[3] Nan Z, Lichao Z, Senlin W, Shifeng W & Yusheng S. (2020) Region-based layered infill area generation of STL models for additive manufacturing. Rapid Prototyping Journal, Vol. 27 No. 1: 99-111.
[4] Munasinghe, N., & Paul, G. (2020) Radial slicing for helical-shaped advanced manufacturing applications. The International Journal of Advanced Manufacturing Technology, 112:1089–1100
[5] King, B., Rennie, A., & Bennett, G. (2020) An efficient triangle mesh slicing algorithm for all topologies in additive manufacturing. The International Journal of Advanced Manufacturing Technology, 112: 1-11
[6] Fu, G., Fu, J., Lin, Z., Shen, H., & Jin, Y. (2016) A polygons boolean operations-based adaptive slicing with sliced data for additive manufacturing. ARCHIVE Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996 (vols 203-210), 231(15) :2783–2799.
[7] Li, H., Wang, T., Sun, J., & Yu, Z. (2016) The adaptive slicing algorithm and its impact on the mechanical property and surface roughness of freeform extrusion parts. Virtual & Physical Prototyping, 11(1): 27-39.
[8] Pan, X., Chen, K., & Chen, D. (2014) Development of rapid prototyping slicing software based on STL model. In: IEEE International Conference on Computer Supported Cooperative Work in Design. IEEE. 191-195
[9] Li, Q., & Xu, X. Y. (2015) Self-adaptive slicing algorithm for 3d printing of fgm components. Materials Research Innovations, 19(suppl.5): S5.635-S5.641.