https://doi.org/10.1007/s41230-021-1090-y

Development of 3D printing entity slicing software

*Yang Guan[1,2], Xun Sun[1,2], Lei Jin[1,2], Xin-li Guo[1,2], Zhi-min Zhang[1], Guo-yan Shui[1,2], Lan-bo Ma[1]

1. Shenyang Research Institute of Foundry Co., Ltd., Shenyang 110022, China

2. State Key Laboratory of Light Alloy Casting Technology for High-end Equipment, Shenyang 110022, China

Abstract: Based on STL (stereo lithography) format file and with Microsoft Visual C^{++} 6.0 programming language, a 3D printing slicing software appropriate for the surface strengthening of the parts has been developed, which includes three functions: 3D model import, model slicing and data export. Through the grouping of STL model triangle facets before slicing, the judgment times of the relationship between the triangle facets and the cutting planes are reduced, and the slicing efficiency is improved. Aiming at the fact that the surface should be strengthened when the part is formed by using of 3D printing process, the function of identifying and marking the surface of the geometric entity is accomplished in the slicing software, which can strengthen the surface of the part according to the user's requirements. The developed slicing software can provide an entity slicing file for 3D printing equipment. The related functions can be adjusted and improved according to user's needs, which makes the software convenient and flexible to use.

Key words: STL format; 3D printing; slicing; surface strengthening

CLC numbers: TP391.9 **Document code:** A **Article ID:** 1672-6421(2021)06-587-06

1 Introduction

With the continuous improvement of product performance, traditional manufacturing methods cannot meet the complex, diversified, rapid and environmental requirements, so, a large number of advanced manufacturing technologies have emerged. One of the advanced manufacturing technologies, additive manufacturing (AM), also known as 3D printing, is the process of depositing material powder layer by layer to form a physical realization of a 3D computer model. It is the opposite of subtractive manufacturing methods [1-2]. AM has found applications in diverse industries such as automotive, aerospace, energy, consumer products and medicine [3-6].

The main processes of 3D printing are the discretization of the model and the stacking of the materials. The discretization of a model is to transform the 3D model into 2D contours, that is, to discretize and transform the 3D model of the geometric entity into the processing codes, which can be recognized by the 3D printer through the slicing software. The stacking of materials is to print the 2D contours layer by layer based on the printing path planning from the slicing data, and finally to form the 3D model. In the process of 3D printing, the 2D closed contour path obtained by slicing determines the movement path of the printer nozzle, and thus determines the quality of the forming part. Therefore, slicing is a key factor that affects the printing quality [7].

Both domestic and overseas scholars have carried out a lot of research work on the slicing process of 3D printing. Wang et al. [8] utilized multi-head slicing algorithms to enhance the speed of 3D printing. Yang et al. [9] accelerated the prefabrication of slicing through parallel slicing algorithms. Kanle et al. [7] improved the efficiency and stability of 3D printing through a slicing and support structure generation algorithm for 3D printing directly on boundary representation (B-rep) models, while Wang et al. [10] reduced the printing time of 3D printing by using of a novel printing direction optimization algorithm. Liu et al. [11] designed an extendable tool path generation software aimed at supporting all 3DP processes. Shin et al. [12] proposed an effective data structure for 3D printing slicer API, which made the adaptation process of 3D printer simpler and more effective. Wasserfall et al. [13] described the implementation of a novel adaptive slicing algorithm for a broad use with common fused deposition modeling (FDM) printers, which can achieve higher printing quality at a reasonably high printing speed. Ahlers et al. [14] increased the printing quality of 3D printing by using of a novel FDM slicing approach. Chen et al. [15] investigated the application of six commercially-available 3D printers by comparing the slicing results of clinically relevant anatomical

***Yang Guan**

Female, born in 1979, Senior Engineer. Research interests: casting process CAD/CAE and additive manufacturing.

E-mail: gy79060977@163.com

Received: 2021-04-28; Accepted: 2021-08-27

models, and obtained the conclusion that the printing cost and time depend on the orientation of anatomy and the selected settings of the printers. In recent years, various 3D printers have been launched abroad, all equipped with self-developed software systems, and some also support general slicing software, such as Cura, Slic3r, Simplify3d, Netfabb, etc. China has also carried out a lot of research work on 3D printing software and gained many achievements. A number of slicing software packages, such as Easy Print 3D, Flash Print, HORI 3D Print, BLT-Slice Viewer, Auto Sector, etc. have been developed independently. But most of the softwares are bundled with their equipment, and the system functions need to be further improved [16]. Aiming at the fact that the surface should be strengthened when the printed green part is formed by using a 3D printing process, a relevant module was developed to ensure dimensional accuracy and to meet the technical requirements of surface quality at the same time, and the corresponding software function module has been realized.

2 Overall design of slicing software

According to the 3D printing process, the slicing software mainly includes three functions: 3D model import, model slicing and data export. The software overall design process is shown in Fig. 1.

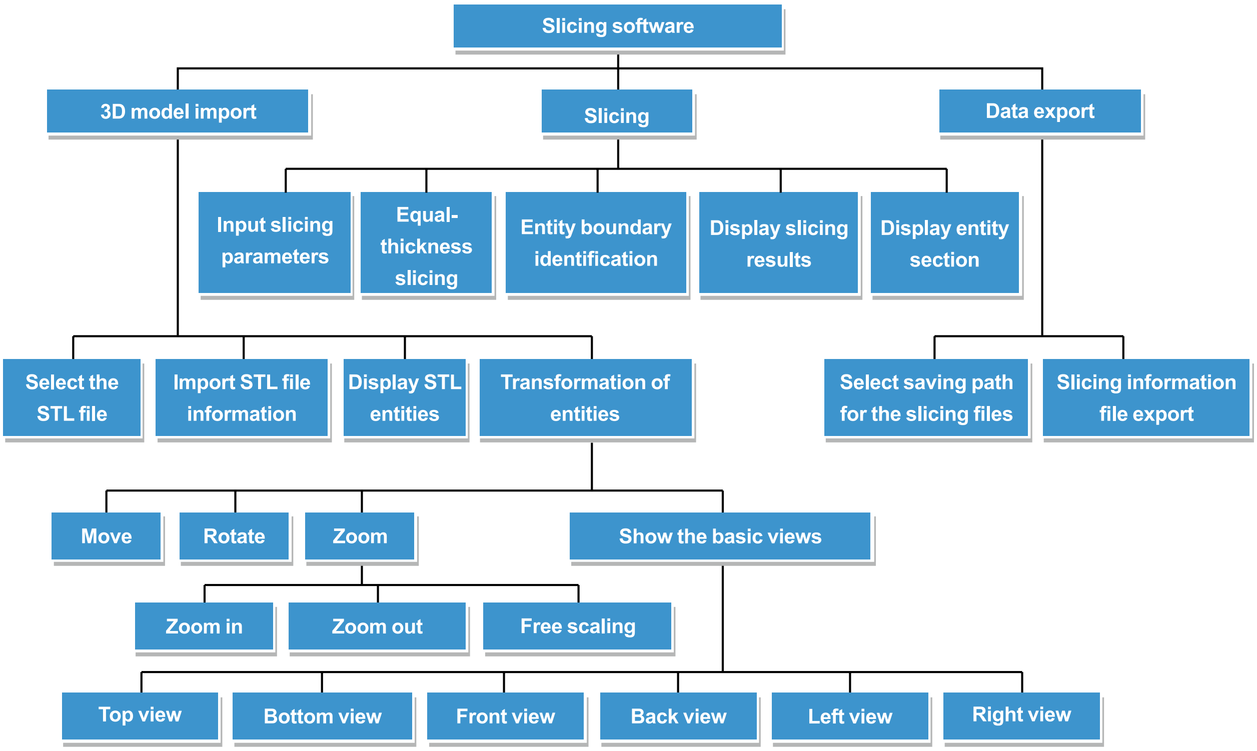


Fig. 1: Flow chart of software overall design

3 Core algorithm and implementation of software functions

3.1 3D model import

3.1.1 File format selection

The layering algorithm of the slicing software can be classified into two categories according to the file format. The first category is the direct layering algorithm based on the CAD 3D model, and the second category is the slicing algorithm based on the STL file of the 3D model. The direct layering algorithm based on the CAD 3D model is not widely used because of its poor compatibility although it has high precision [17]. STL is the de facto standard of the most current 3D printing processes and machines [18]. STL file format is an intermediate file format developed in 1988 in reference to the idea of finite element method mesh generation by Albert Consulting Group of 3D System Company in USA [19]. It uses a series of small plane triangle facets to fit a 3D model surface, and its format is simple and clear, convenient and swift, easy to be accepted and understood. STL format file is the most mature and widely used slicing data source in the industrial application of CAD/CAE/RP [20]. Therefore, STL file is selected as the imported 3D model file format in this study.

3.1.2 Implementation of 3D model import

The 3D model import module recognizes the functions of selecting and opening the STL file of 3D printing geometric entities, importing STL file information, displaying 3D geometric entities (Fig. 2), and realizing the translation, rotation, zoom and display of basic views of 3D geometric entities.

3.2 Model slicing

3.2.1 Core algorithm

No matter which kind of format the 3D model adopted, the essence of slicing is to obtain the contour data of *X*-*Y* cross section by iteratively cutting the model in the *Z* direction using certain thickness. Different data formats have different slicing methods for 3D models, but the core of all the algorithms is to adopt the mathematical idea of achieving the intersections of the geometric model and slicing planes. At present, the slicing methods of the model are divided into two classes according to the thickness of slices, one is equal-thickness slicing, the other is self-adaptive layer thickness slicing [21]. The equal-thickness slicing is to keep the layer thickness constant during the whole slicing process. The advantage of this algorithm is simple and fast, however, in the case of thicker layers, there is an obvious step effect between the two adjacent layers, and the precision of the generated model is low. According to the requirements of users, the thickness of each layer in the self-adaptive layer slicing should be established on the basis of the previous layer. Although the precision of the generated model is high, the algorithm is difficult to realize, and the calculation speed is slow. If there is no special requirement, the equal-thickness slicing can satisfy the precision requirement of a 3D printing model. For the standard of high precision, the requirement of thinning the layers can be met in the algorithm of equal-thickness slicing.

In this study, the algorithm of equal-thickness slicing is selected, and the flow chart is shown in Fig. 3.

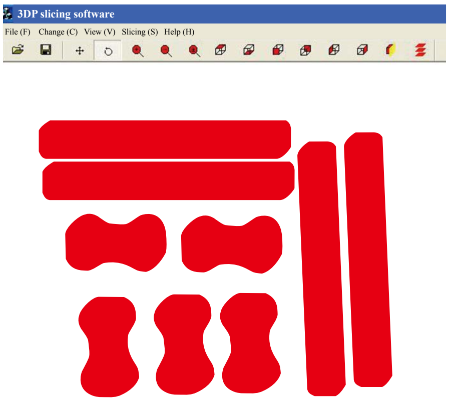


Fig. 2: Displaying 3D geometric entities

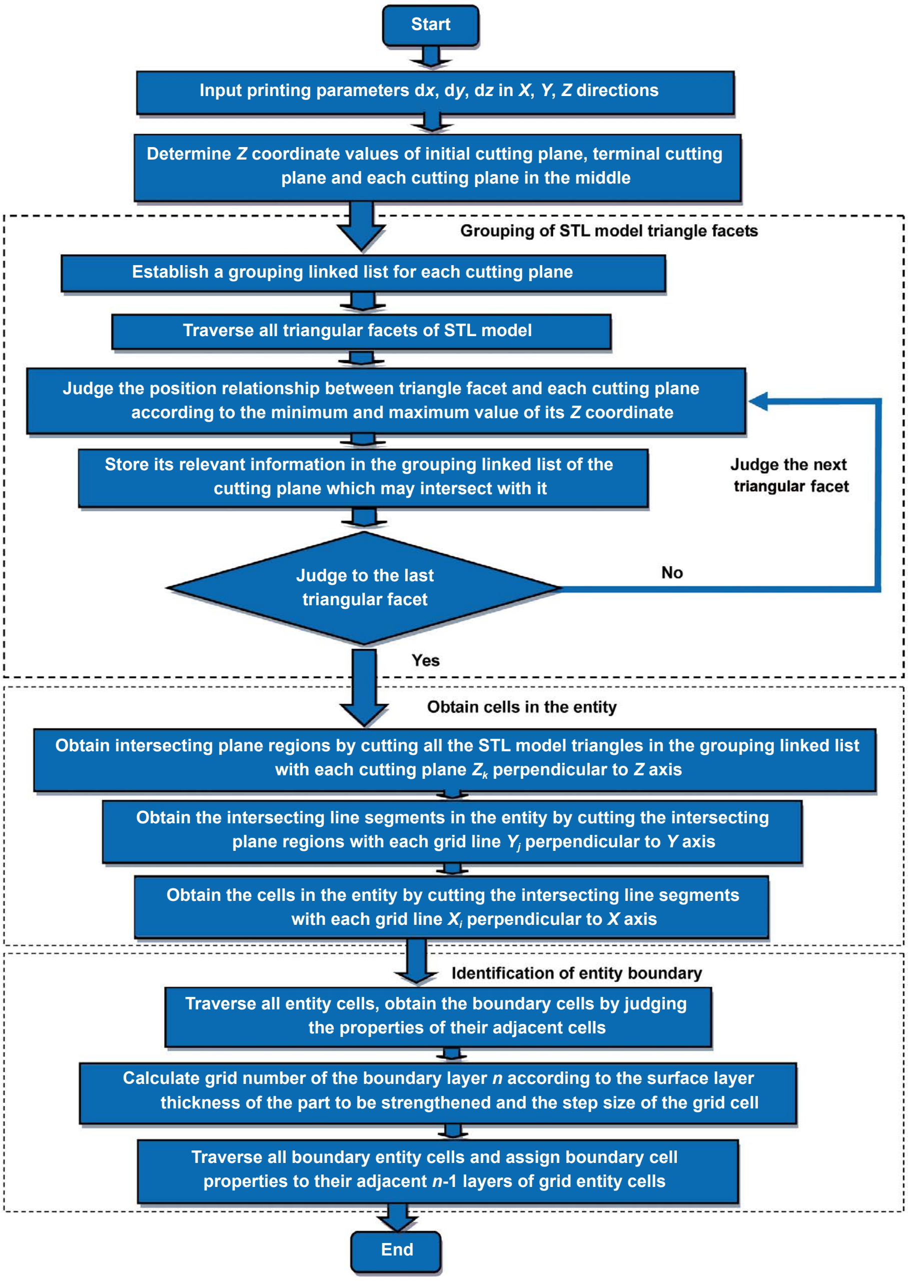

Start
Input printing parameters dx, dy, dz in X, Y, Z directions
Determine Z coordinate values of initial cutting plane, terminal cutting plane and each cutting plane in the middle
Grouping of STL model triangle facets
Establish a grouping linked list for each cutting plane
Traverse all triangular facets of STL model
Judge the position relationship between triangle facet and each cutting plane according to the minimum and maximum value of its Z coordinate
Store its relevant information in the grouping linked list of the cutting plane which may intersect with it
Judge the next triangular facet
Judge to the last triangular facet
No
Yes
Obtain cells in the entity
Obtain intersecting plane regions by cutting all the STL model triangles in the grouping linked list with each cutting plane Z_k perpendicular to Z axis
Obtain the intersecting line segments in the entity by cutting the intersecting plane regions with each grid line Y_j perpendicular to Y axis
Obtain the cells in the entity by cutting the intersecting line segments with each grid line X_i perpendicular to X axis
Identification of entity boundary
Traverse all entity cells, obtain the boundary cells by judging the properties of their adjacent cells
Calculate grid number of the boundary layer n according to the surface layer thickness of the part to be strengthened and the step size of the grid cell
Traverse all boundary entity cells and assign boundary cell properties to their adjacent n-1 layers of grid entity cells
End

Fig. 3: Flow chart of equal-thickness slicing algorithm

The slicing process includes 3D model slicing and printing path planning. The 3D model slicing is to obtain the intersecting plane region of the cutting planes and the STL entity, and the result is a series of closed contour loops which are made up of intersecting line segments connected end to end. After slicing the 3D model, it is necessary to generate scanning filling lines by planning the printing paths in the contour loops; the scanning filling lines are the processing paths of the 3D printer for each layer of the entity. At present, the most basic path planning algorithms are divided into parallel line scanning, self-adaptive parallel line scanning, contour offset scanning, partition scanning, fractal scanning, layered direction changed scanning, etc. Among them, the parallel line scanning algorithm uses relatively simple line segments to fill the whole contour area in parallel from top to bottom or from bottom to top. The self-adaptive parallel line scanning algorithm uses adaptive parallel lines to fill the contour, that is, filling lines with smaller width are used where the curvature of the contour changes greatly, and filling lines with larger width are used where the curvature changes less. The contour offset scanning algorithm offsets the overall contour ring to the entity direction for a certain distance, and repeats it many times until the whole entity contour area is filled. The partition scanning algorithm divides the cross-section of the 3D model into many small areas, and fills the small areas with parallel line scanning pattern. The fractal scanning algorithm uses the Hilbert space curve filling method in the principle of computer graphics. The layered direction changed scanning algorithm is improved based on the partition scanning algorithm, which means that the scanning lines of two adjacent layers are inconsistent in direction [22].

This study uses the parallel line scanning algorithm, which is the earliest proposed and most mature path planning method. It is also the most basic path planning method adopted by most slicing software packages at present. Its basic idea is to scan each side of the intersecting plane region of the cutting planes and the STL entity using horizontal lines parallel to the *X*-axis from negative to positive of the *Y*-axis, and obtaining the intersection points of each scanning line and the intersecting plane region boundary. After removing redundant intersection points, the intersection points are sorted, and then connected into line segments in turn, which are the paths taken by the nozzle of the 3D printer.

The implementation steps of the slicing algorithm are as follows:

(1) Choose *Z* axis as the slicing direction, and set the slicing thickness to d*z*.

(2) Input printing parameters d*x* (step length along *X* axis), d*y* (step length along *Y* axis), d*z* (slicing thickness) in *X*, *Y*, *Z* directions by users.

(3) Determine the *Z* coordinate values of initial cutting plane, terminal cutting plane, and each cutting plane in the middle according to the minimum value $z_{\min}$ and the maximum value $z_{\max}$ of the STL model in the slicing direction, that is, the *Z* direction.

(4) Group the STL model triangle facets.

(a) Establish a grouping linked list for each cutting plane to store the information of all STL model triangles that may intersect with it.

(b) Judge the position relationship between each triangular facet in the STL model and each cutting plane according to the minimum and maximum value of its *Z* coordinate. The relevant information is stored in the grouping linked list of the cutting plane which may intersect with it.

(c) Traverse all triangular facets of STL model, then the grouping of STL model triangle facets will be completed.

(5) Cut all the STL model triangles in the grouping linked list with each cutting plane perpendicular to the *Z* axis (Z_k) from the initial cutting plane, and increasing one slicing thickness *dz* each time, obtaining the intersecting plane regions of the cutting planes and the STL entity, until the terminal cutting plane.

(6) Cut the intersecting plane regions with each grid line perpendicular to the *Y* axis (Y_j) from the minimum value $y_{\min}$ of the STL model in *Y* direction, and increasing a grid step dy each time, obtain the intersecting line segments in the entity, until the maximum value $y_{\max}$ of the STL model in *Y* direction. Repeat for each intersecting plane region.

(7) Cut the intersecting line segments with each grid line perpendicular to the *X* axis (X_i) from the minimum value $x_{\min}$ of the STL model in *X* direction, and increasing a grid step d*x* each time, obtain the cells in the entity, until the maximum value $x_{\max}$ of the STL model in *X* direction. Repeat for each intersecting line segment.

(8) Identify entity boundary.

(a) Obtain the properties (entity or non-entity) of all grid cells in the calculation domain, then traverse all entity cells and judge the properties of their adjacent cells. If all are entity, the cell is not a boundary cell; Otherwise, the cell is a boundary cell, and the boundary cell property is assigned to it.

(b) Calculate the grid number of the boundary layer n = int (t/d) according to the surface layer thickness of the workpiece to be strengthened, determined by the user t and the step size of the grid cell d.

(c) Traverse all boundary entity cells and assign boundary cell properties to their adjacent n-1 layers of grid entity cells.

3.2.2 Key problems in slicing process and their solutions

(1) Group STL model triangle facets

During the slicing process, because the triangle facets of STL model are scattered and disorderly, and there is no topological correlation information between any two triangle facets, it is necessary to judge the position relationships between all the triangle facets and the cutting plane every time when the intersecting plane region of the cutting planes and the STL entity is obtained, which is time-consuming. To realize fast and efficient slicing, the key to the slicing algorithm is to reduce the judgment times of the relationships between the positions of the triangle facets and the cutting planes. Therefore, the coordinate information of the triangle facets is used to group in advance, so as to improve the efficiency of slicing.

(2) Special cases handling for scanning lines filling

During the calculation of the intersection points of scanning lines and the intersecting plane regions, when a scanning line intersects the intersecting plane polygon at a vertex or coincides with an edge of the polygon, as shown in Fig. 4, a wrong scanning path will be obtained. To avoid this situation, the scanning line will be slightly offset through increasing the *Y* coordinate value of the scanning line by 0.1 μm.

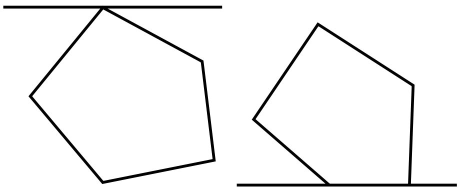


Fig. 4: A scanning line intersects the intersecting plane polygon at a vertex or coincides with an edge of polygon

(3) Identification of entity boundary

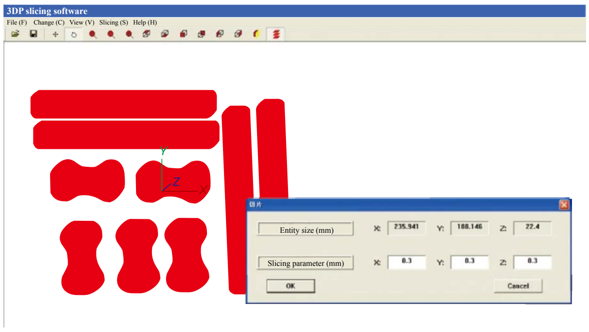
In the 3D printing process, sometimes it is necessary to strengthen the surface of the printing green part to improve the surface quality. Therefore, the function of identifying and marking the surface of the geometric entity is needed in the slicing software. This function is realized in the developed slicing software as discussed in the Section 3.2.1.

3.2.3 Implementation of slicing function

The slicing module realizes the functions of inputting slicing parameters (Fig. 5), equal-thickness slicing, entity boundary identification, displaying entity slicing results [Fig. 6(a)] and displaying entity section [Fig. 6(b)]. As can be seen from Fig. 6(b), the boundary cells (black part) of the entity is identified, and the user can strengthen the entity surface accordingly.

3.3 Data export

After filling each layer of the profiles of the geometric entity with a scanning strategy, the data information of the entire layer needs to be output as a data file or control file that can be recognized by a 3D printing device. In this study, the information of the slicing cells of the entity is saved in files as "00", "01" and "11", among which "00" represents non-entity cells, "01" represents inner entity cells and "11" represents boundary entity cells. The slicing files can be saved in both ASCII and binary formats. The data export module exports the slicing file to the path specified by the users.



Entity size (mm) X: 235.941 Y: 188.146 Z: 22.4

Slicing parameter (mm) X: 0.3 Y: 0.3 Z: 0.3

Fig. 5: Inputting slicing parameters in *X*, *Y*, *Z* directions

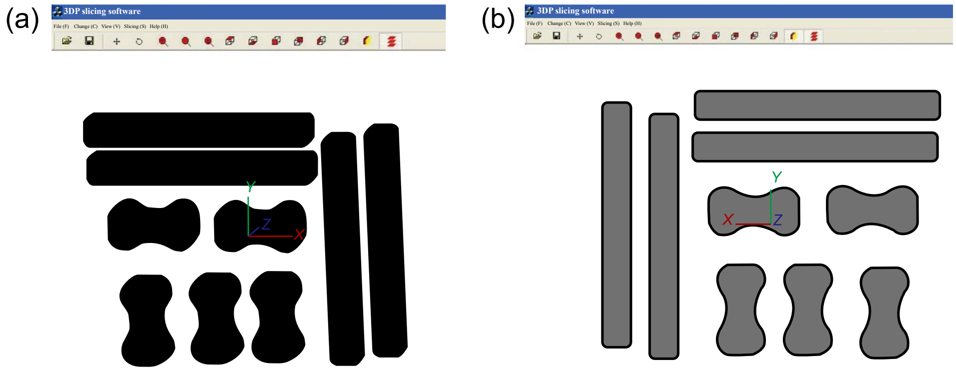

(a)
3DP slicing software
File (F) Change (C) View (V) Slicing (S) Help (H)
Y
Z
X
(b)
3DP slicing software
File (F) Change (C) View (V) Slicing (S) Help (H)
Y
X
Z

Fig. 6: Displaying slicing results (a) and sections (b) of geometric entities

3.4 Case verification

The practicality and accuracy of the software is verified by comparing the appearance of the entity before and after slicing, as well as the section of the entity after slicing. The entity comparison of tetrahedron and circle before and after slicing is shown in Figs. 7–8. The comparison results show that the sliced entities are consistent with the STL entities.

4 Conclusions

(1) The discretization of 3D printing geometric entity is realized by adopting the slicing algorithm based on the STL format file. The efficiency of slicing is improved by grouping the triangle facets of the STL model in advance. The special cases in which a scanning line intersects the intersecting plane polygon at a vertex or coincides with an edge of the polygon are avoided through the slight offset of scanning lines. By identifying and marking the entity boundary cells and coordinating with the process parameter adjustment of the printer, the function of strengthening the surface of the printing part can be achieved.

(2) Three main functions of 3D model import, model slicing, and data export are realized in the software.

(3) The practicality and accuracy of the developed software are verified by slicing the typical geometric entities.

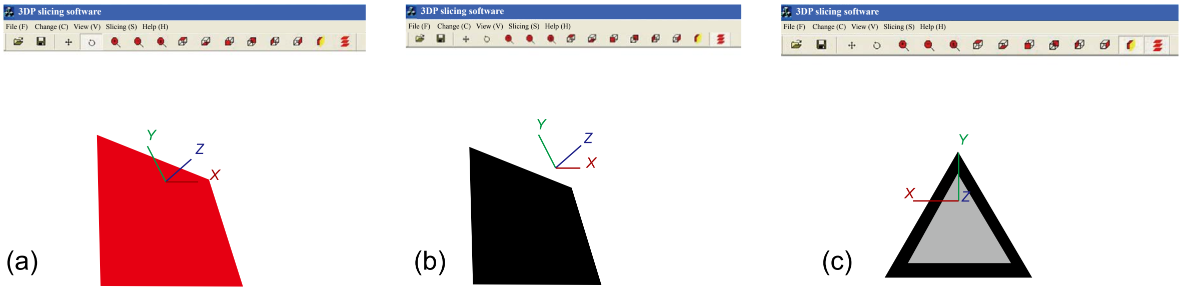


Fig. 7: Comparison of tetrahedron before and after slicing: (a) STL entity; (b) entity after slicing; (c) section of the entity after slicing

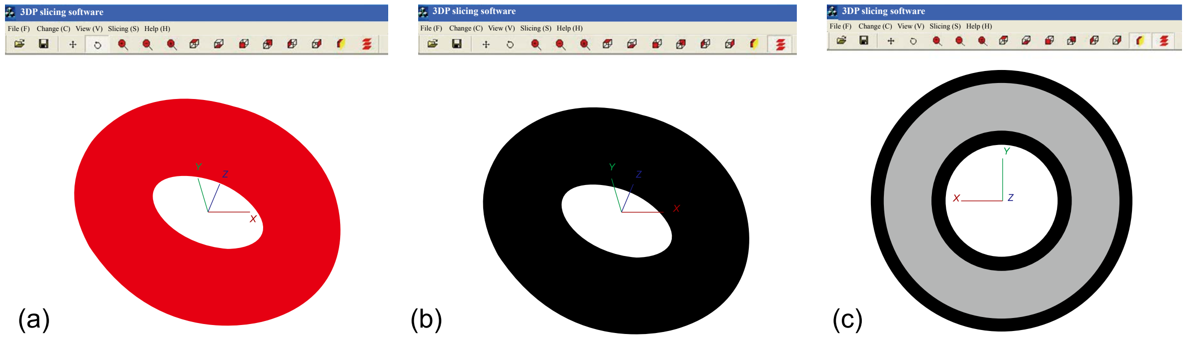


Fig. 8: Comparison of ring before and after slicing: (a) STL entity; (b) entity after slicing; (c) section of the entity after slicing

References

[1] Munasinghe N, Paul G. Radial slicing for helical-shaped advanced manufacturing applications. The International Journal of Advanced Manufacturing Technology, 2021, 112: 1089–1100.

[2] Ngo T D, Kashani A, Imbalzano G, et al. Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. Compositions, Part B, 2018, 143: 172–196.

[3] Shi Y S, Zhang J L, Wen S F, et al. Additive manufacturing and foundry innovation. China Foundry, 2021, 18(4): 286–295.

[4] Tino R, Moore R, Antoline S et al. COVID-19 and the role of 3D printing in medicine. 3D Print in Medicine, 2020, 6(11): 1–8.

[5] Camacho D D, Clayton P, O'Brien W, et al. Applications of additive manufacturing in the construction industry. Automation in Construction, 2018, 89: 110–119.

[6] Cheng G Z, Folch E, Wilson A, et al. 3D printing and personalized airway stents. Pulmonary Therapy, 2017, 3(1): 59–66.

[7] Shi K L, Cai C H, Wu Z J, et al. Slicing and support structure generation for 3D printing directly on B-rep models. Visual Computing for Industry, Biomedicine, and Art, 2020, 2: 3.

[8] Wang Y X, Gu Z Q, Song L, et al. Speeding up 3D printing using multi-head slicing algorithms. In: Proc. 5th International Conference on Enterprise Systems, Beijing China, 2017: 99–106.

[9] Yang J M, Chen Y, Huang W D, et al. Survey on artificial intelligence for additive manufacturing. In: Proceedings of the 23rd International Conference on Automation & Computing, 2017, doi:10.23919/IConAC.2017.8082053.

[10] Wang W M, Shao H L, Liu X P, et al. Printing direction optimization through slice number and support minimization. IEEE Access, 2020(8): 75646–75655.

[11] Liu J, Fan Y B, Lu Q H. Design of extendable tool path generation software for 3D printing. In: Proc. International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO), 2014: 130–133.

[12] Shin H S, Lee H I, Jang E S. An effective data structure for a 3D printing slicer API. In: Proc. International Conference on Consumer Electronics - Asia (ICCE-Asia), IEEE, 2016.

[13] Wasserfall F, Hendrich N, Zhang J W. Adaptive slicing for the FDM process revisited. In: Proc. 13th IEEE Conference on Automation Science and Engineering (CASE), 2017: 49–54.

[14] Ahlers D, Wasserfall F, Hendrich N, et al. 3D printing of nonplanar layers for smooth surface generation. In: Proc. 15th International Conference on Automation Science and Engineering (CASE), IEEE, 2019: 1737–1743.

[15] Chen J V, Dang A B C, Dang A. Comparing cost and print time estimates for six commercially-available 3D printers obtained through slicing software for clinically relevant anatomical models. 3D Printing in Medicine, 2021, 7: 1–14.

[16] Heng N R. Research on control system and slicing algorithm for wire arc 3D printer. Master dissertation, Chongqing: Chongqing University, 2018. (In Chinese)

[17] Choi S H, Kwok K T. Hierarchical slice contours for layered manufacturing. Computers in Industry, 2002, 48(3): 219–239.

[18] Oropallo W, Pieg L A. Ten challenges in 3D printing. Engineering with Computers, 2016, 32: 135–148.

[19] Kai C C, Gan J, Du Z, et al. Data structure in rapid prototyping and manufacturing. California: Database and Data Communication Network System, 2002: 367–416.

[20] Sabourin E, Houser S A, Bøhn J H. Adaptive slicing using step wise uniform refinement. Rapid Prototyping Journal, 1996, 2(4): 20–26.

[21] Hu J. Study on STL-based slicing process for 3D printing. In: Proceedings of the 28th Annual International Solid Free Form Fabrication Symposium, 2017: 885–895.

[22] Fan W B. Research and development of slicing software for SLM metal forming equipment. Dissertation, Shanxi: North University of China, 2019.