Research on the Perforating Algorithm Based on STL Files

Han Yuchuan¹,*, Zhu Xianfeng¹, Bai Yunrui¹ and Wu Zhiwen¹
¹College of Precision Instrument and Opto-electronics Engineering, TianJin University, TianJin 300072

*han_yuchuan2012@163.com

Abstract. In the process of making medical personalized external fixation brace, the 3D data file should be perforated to increase the air permeability and reduce the weight. In this paper, a perforating algorithm for 3D STL file is proposed, which can perforate holes, hollow characters and engrave decorative patterns on STL files. The perforating process is composed of three steps. Firstly, make the imaginary space surface intersect with the STL model, and reconstruct triangles at the intersection. Secondly, delete the triangular facets inside the space surface and make a hole on the STL model. Thirdly, triangulate the inner surface of the hole, and thus realize the perforating. Choose the simple space equations such as cylindrical and rectangular prism equations as perforating equations can perforate round holes and rectangular holes. Through the combination of different holes, lettering, perforating decorative patterns and other perforated results can be accomplished. At last, an external fixation brace and an individual pen container were perforated holes using the algorithm, and the expected results were reached, which proved the algorithm is feasible.

1. Introduction

STL (Stereo lithographic) is an interface file of the three-dimensional solid modelling system proposed by 3D Systems Company in 1987, which has become the most commonly used 3D data format and the actual interface standard in the field of rapid prototyping presently [1]. The STL file approximates the real 3D model with small triangular facets, and has the characteristics of controllable precision, format conversion conveniently and segmentation easily. The STL file has been widely used in rapid prototyping, displaying of image real sense, finite element analysis, and reverse engineering [2].

Because the STL file format is simple and universal, it is supported by almost all platforms, and most of the three-dimensional modelling systems’ interface data is saved as STL file format, such as 3D scanning data and 3D models produced by modelling software. However, due to the lack of topological information for the small triangular facets and the loss of precision of real models [3], the STL files are only used for displaying and translating, which means it is hard to be edited [4-6]. In order to meet the need of ventilation, individuation, saving material and reducing weight in some special scenes, this paper proposes a perforating algorithm based on STL files. The algorithm can help perforating round holes, elliptical holes, square holes and triangular holes, and through the combination of a number of holes, the complex structure of holes can be perforated, including character holes, pattern holes, etc.

2. The STL file’s format and its specification
The STL file uses a lot of triangular facets to represent the 3D object, and each triangular facet includes two parts: vertex coordinates and the normal vector [7]. The process of representing a cube as the STL file is shown in figure 1.

**Figure 1.** Triangulation of a cube

The STL file has two formats: ASCII code and binary. ASCII code file is readable, while binary file takes up less memory, which is easy to read and store. The perforating process of STL files needs to do various operations on a large number of triangular facets, and speed is a very important factor, so this paper selected binary file as the research object. Its format is shown in table 1[8]:

| Data type     | Storage information | The number of triangle facets | The number of triangle facets(50 bytes) | ... |
|--------------|---------------------|-------------------------------|----------------------------------------|-----|
| Bytes        | Text information    | Normal vector                 | Vertex 1 | Vertex 2 | Vertex 3 | Attribute byte | ... |
| 80           | Unsigned char       | 4                             | 3×4      | 3×4      | 3×4      | 2              | ... |
| 4            | Unsigned long integer | Float | Float | Float | Float | Unsigned short integer | ... |

The STL file is a collection of triangular facets, and each triangular facet is represented by one normal vector and three coordinates of the three vertices. The STL file must follow the following rules [9]: (1) Common-vertex rule. Each triangle must share two vertices with adjacent triangles, which means the vertices of a triangular facet cannot lie in the edges of adjacent triangles. (2) Orientation rule. The normal vector of each triangular facet and the three edges of the triangle satisfy the right-hand rule, and the normal vector direction points to the outside of the model. (3) Full rule. Triangular facets must cover all the outer surfaces of the model, without any omission. (4) Value rule. The coordinates of each vertex of the STL file must be nonnegative, i.e., the model must be in the first quadrant of the coordinate system.

3. Basic thought
Use closed space surfaces (such as hollowed cylindrical surface) to intersect with the STL file, delete the STL files inside the closed space surface, reserve the STL files outside the closed space surface, reconstruct triangles on the space closed surface to form the inner surface of the hole, and then complete perforating.

Therefore, the perforating processes are as follows:
1). Construct the imaginary closed space surface mathematical equation whose image in the Descartes coordinate system is a closed space surface, which intersects with the STL file.
2). Calculate the crossover points of the closed space surface and the STL file, reconstruct triangles by the crossover points and the vertices of the triangular facets at the intersection, delete the original triangular facets so that all the triangular facets are totally inside and outside the closed space surface.
3). Delete the triangular facets inside the closed space and form a hole on the STL file.
4). Using the triangular facets to fill the inner surface of the hole, and the perforating process is realized.

4. Perforate round holes

4.1. The crossover points of a hollowed circular cylinder and a sphere
For the convenience of research, this paper chose a spherical STL file as the research object, and the perforating hollowed cylindrical equation is given by

$$\left(\bar{z} - z\right)^2 + \left(\bar{x} - x\right)^2 = r^2$$

This equation describes a y-infinity cylindrical surface, where \(\bar{z}\) is the mean value in the z-direction of the points on the cylindrical surface, \(\bar{x}\) is the mean value in the x-direction of the points on the cylindrical surface, \(r\) is the radius of the cylinder [10]. The intersection of the hollowed circular cylindrical and the sphere is shown in figure.2.

![Figure 2](image)

**Figure.2** Intersection of one hollowed circular cylinder and a sphere

The STL file consists of numerous triangular facets, each of which has three vertices, and each vertex corresponds to a space coordinate \((x, y, z)\) [11]. The position relation between the triangular facet and the hollowed circular cylinder can be found by judging the position relation between the three vertexes’ coordinates and the hollowed circular cylinder. By searching for the number of vertices of each triangular facet in the outer and inner surface of the hollowed circular cylinder, the intersecting edges of the triangle can be found, seeing table.2.

| case  | inner | outer | surface | intersecting lines | Intersecting situations |
|-------|-------|-------|---------|--------------------|------------------------|
| 1     | 2     | 0     | 2       | inner-outer1, inner-outer2 |
| 2     | 1     | 0     | 2       | inner1-outer, inner2-outer |
| 3     | 1     | 1     | 1       | inner-outer |
| others| m     | n     | p       | none |

After knowing the intersecting lines, the crossover points can be obtained by the equation of the line intersecting with the surface.

The parameter equation of the intersecting line is [12]:

$$x = x_0 + t \cdot l$$
$$y = y_0 + t \cdot m$$
$$z = z_0 + t \cdot n$$

(2)
Where \((x_0, y_0, z_0)\) is a known point of the intersecting line, and \((l,m,n)\) is the direction vector of the line. By taking (2) into (1), we can get \(t\), and then get the crossover point.

4.2. Reconstructing triangles at intersecting lines
Reconstruct triangles with crossover points and the corresponding triangles’ vertexes to form a circular contour on the surface of the sphere, to help to perforate a circular hole. There are five kinds of intersections between the surface and the triangles [13]. 1. Two edges intersect with the surface. 2. One edge intersects with the surface as well as one point is located on the surface. 3. One edge is located on the surface. 4. One point is located on the surface and the other two points are located outside the surface. 5. The three vertexes are located outside the surface. See figure 3.

When reconstruct a triangle, there is no need to deal with the case 3, 4, and 5. For case 1, the two crossover points construct a triangle with a single vertex on one side of the surface, and the two crossover points construct two triangles with the other two vertices. For case 2, the crossover point, the interior point, and the upper and lower separate vertices construct two triangles. After the constructions are finished, completed triangles can be formed on both sides of the surfaces, seeing figure 4.

![Figure 3. The five situations of triangles and a surface](image)

![Figure 4. Reconstruct triangles](image)

The result of reconstructing triangles at the intersection of the sphere and the hollowed circular cylinder mentioned above is shown in figure 5.

With the hollowed circular cylinder as the boundary, delete the triangular facets inside the hollowed circle cylinder, and a sphere with a circular hole is formed, as is shown in figure 6.

![Figure 5. Reconstructing triangles at the intersection](image)

![Figure 6. The spherical with a circular hole](image)

4.3. Clockwise sorting of boundary points
According to the common-vertex rule and full rule of STL files, each edge of the triangle has only two triangles sharing [14]. The STL file shown in figure 6 is incomplete, which has to be filled with triangular facets inner the hole. If an edge has only one triangular sharing, the edge is the boundary. According to this feature, we can find the boundary edges, and then find the boundary points [15, 16].

The boundary points are unordered, and if using the boundary points of the two ends constructs triangles in the inner surface directly, intersections and holes will occur. To construct the complete
inner surface, we need to sort the boundary points, and then can we use the top and bottom boundary points to construct the triangles sequentially [17].

Clockwise sorting method is used here to sort the boundary points based on its tangent value. Project the boundary points to the x-z plane for reducing dimension, and then sort the boundary points according to their tangent value on the x-z plane. First, calculate the centre of the boundary points. Second, reconstruct an x-z coordinate axis with the centre point as the origin, and divide the boundary points into the left and right sides of the z-axis. Third, sort the boundary points of each side by the tangent value. Forth, connect the two-part points to achieve a clockwise sorting.

For the point on the z-axis, if it is on the top, divide it into the right boundary point ring, and if it is at the bottom, divide it into the left boundary point ring. Thus avoid the denominator of zero, and then achieve the sorting, seeing figure 7.

**Figure 7.** Tangent sorting

### 4.4. Inserting points in the way of interpolation

If the number of points in the top and bottom boundary is the same, construct triangles with the top and bottom points directly to form the inner surface of the hole. Otherwise, it is necessary to insert points in the fewer points’ boundary, or the inner surface will not be smooth. When inserting points, first calculate the difference of the number of the top and bottom points, and set it as P, which means we need to insert P points to the fewer points’ boundary. Set the number of its boundary points as W, insert the first point between the first and second boundary points, and insert the second point between the second and third boundary points... If P is not bigger than W, insert P points in this way. If P is bigger than W, update P’s value as P-W after traversing a circle, and then iterate back until P points are inserted. Add each inserted point to the back of the W, and then sort the points again after all the insertions are completed.

Assume that the first point is being inserted (set it as point P). Set the coordinates of the first point and the second point on the boundary points as \((x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\). Set the inserted point’s x-value as \(\bar{x} = \frac{x_1 + x_2}{2}\), and the value of \(z\) can be calculated by \(z = \bar{z} + (x - \bar{x}) = r\). Knowing the x-value and z-value, we can find out the point’s projection on x-z plane fallen in which triangular facet of the sphere STL file. If point P satisfies the formula (3) and (4), it is in triangle ABC.

\[
BP = u \times BA + v \times BC
\]

\[
u \geq 0, v \geq 0, u + v \leq 1
\]

The u and v are the coefficients, and the schematic is as figure 8.

**Figure 8.** The judgment of whether one point is in a triangle
This triangle is the projection of the space triangle on the x-z plane. By traversing the sphere STL file in a certain range, we can find the triangular facet satisfied with the formula (3) and (4). The triangle facet is corresponding to the point P. Then obtain the crossover point of the triangular facet with the straight line parallel to the y-axis as well as goes through the point P. The equation of the triangular facet is:

\[
AA \times (x - A(1,1)) + BB \times (y - A(1,2)) + CC \times (z - A(1,3)) = 0
\]

AA, BB, CC are the three components of the triangle ABC’s normal vector. A(1,1), A(1,2), A(1,3) are the three components of point A. By bringing \( x = x_p \) and \( z = z_p \) into the equation (1), the value of \( y \) can be obtained as \( y_p \), and the \((x_p, y_p, z_p)\) is just the coordinate of the point P.

When the number of the top and bottom boundary points is the same, the inner surface of the cylinder can be constructed using the top and bottom boundary points in order. See figure 9.

**Figure 9.** The process of constructing the inner surface

Combine the inner surface of the cylinder with the outer triangle facets of the cylinder on the sphere can realize the perforating, seeing figure 10.

If the cylindrical equation is changed into elliptic cylinder equation, the elliptic hole can be perforated, as is shown in figure 11.

**Figure 10.** Perforate a round hole

**Figure 11.** Perforate an elliptic hole

5. Perforate polygon holes
For a space polygon prism that cannot be represented by a single equation, the method proposed above cannot be used for perforating polygon holes. At this time, the space polygon equation can be decomposed into several surface equations, and each surface equation can form a surface in the Descartes coordinate system, which can be used as one of the inner surfaces of the hole. Combine the
inner surface of the hole with the outer triangle facets of the polygon prism on the sphere can realize the perforating.

5.1. Set vertices
For the convenience of research, this paper chose the z-infinity square prism as the perforating model, and selected the sphere mentioned above as the perforated object. First of all, set the four vertices as the projection of the cuboid’s prism on the x-y plane, which were (200,200), (200,300), (300,200), (300,300) here.

Obtain the crossover points of the square prism’s edges with the triangular facets of the STL file. In order to find the triangular facets intersected with the edges, it is necessary to judge each of the projected points falls into which triangle. After the triangle is found, the crossover point of the edge and the triangle can be obtained. Use the crossover point and the three vertices of the triangle reconstruct three triangles and delete the original triangle, as shown in figure 12. This allows the triangles at the vertices of the square to be smoothly divided into small, complete triangles. Thus, STL files can be manipulated directly when cutting with a square prism without having to deal with the complex situation that the prism’s edge is located in a triangle.

![Figure 12. Reconstruction of triangles at the square’s prisms](image)

![Figure 13. The sphere with a square hole](image)

5.2. Intersections of planes and the sphere
Using line segments AB, BC, CD, and DA to divide the triangular facets intersected with them separately. Delete the triangles facets inside the square prism, and the rest of the triangle facets of the STL file can form a sphere with a square hole, see figure 13.

5.3. Triangulate inner surfaces
Find the top and bottom boundary points of the STL file shown in Figure 13. Each side’s boundary points are distributed on four edges. Sort the points in each edge. If the number of points in the two boundaries is the same, construct triangles directly, otherwise, we need to insert points to make them the same. Construct triangles with the corresponding top and bottom boundary points, which means complete the inner surface of the hole, and together with the outer STL files of the square prism can realize perforating, as is shown in the figure 14.

![Figure 14. Perforate a cube hole](image)

![Figure 15. Perforate a triangle hole](image)
When the square prism equation is changed into a triangular prism equation, the triangular hole can be perforated, as is shown in figure 15.

6. Examples and analysis
Through coordinate translation and matrix transformation, it is possible to perforate holes of multiple directions with different shapes on the STL file. The rotation matrices in the x-direction, y-direction, and z-direction are as follows [18]:

\[
R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \quad R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad R_z(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

6.1. Individual external fixation brace
Figure 16 shows the STL model of a wrist personalized external fixation brace, which was generated by offsetting the triangle facets of a scanned wrist STL file. In order to make the model breathable and light, it needs to be perforated. By perforating several round holes, the perforating effect can be realized, as is shown in figure 17.

![Figure 16. The wrist’s personalized external fixation brace](image)

![Figure 17. Overall perforating effect](image)

After 3D printing out the individual external fixation brace, it can be worn on the arm, as is shown in figure 18. It can be seen that these holes can make the brace breathable, which can prevent the side effects such as swelter and itching in the treatment of fracture, meanwhile the hole can lighten the whole weight of the brace and reduce the burden for the patient.

![Figure 18. Perforated individual external fixation brace](image)

6.2. Pen Container
With the rapid development of 3D printing technology, 3D printing has gradually entered the homes of ordinary people. 3D model websites have many 3D models that can be downloaded, printed, and used directly, such as architectural models, toys, tools and so on. However, generally the models’ format is the STL file, which is inconvenient to be edited. The figure 19 shows a pen container model downloaded from 3D tiger website, which can be used as a pen container as well as a mobile phone holder.
However, the pen container’s shell has no patterns, which is slightly dull, and will consume much material when it is printed. At this time, it can be perforated by the perforating algorithm proposed above. Through perforating different text and patterns can achieve personalized perforating effort so that it becomes one’s own personalized pen container. By using different spatial equations such as circle, ellipse, rectangle, and triangle to perforate the STL file, meanwhile control the drilling depth so that it does not run through the whole model, we can achieve the perforating effect as shown in figure 20.

7. Conclusion
Formed 3D data is generally STL file format, the sources of which are 3D scanner, 3D model designing software, etc. This format file is easy to be transmitted and displayed, but not easy to be edited. In this paper, a perforating algorithm based on STL file is proposed, which can be used to perforate the three-dimensional model, to meet the special requirements of ventilation, reducing weight and so on. Aiming at perforating mesh files, this paper proposed a method of spatial surface cutting. The crossover points of the cutting surface and the STL file are sorted by the tangent value, and then the inner surface is triangulated, that is, the perforating is completed. Through the coordinate transformation and matrix transformation, a plurality of different holes can be perforated on an STL file, which means complete the whole perforating. This paper verified the effect of the whole perforating with the personalized external fixation brace and the personalized pencil container, and proved the feasibility and practicability of the algorithm. Nevertheless, this paper’s perforating pattern is just basic pattern of the circular, square, and other simple geometric shapes, and if we can establish the complex pattern (such as Chinese traditional auspicious clouds pattern) equation’s 3D mathematics, the corresponding perforating operation can also be achieved. This is the future research direction of this topic.

Acknowledgments
At this time I want to thanks to my tutor—Zhu Xianfeng, who gave me many academic and constructive advices in composing this paper. Besides, he also helped me in other aspects such as reading and thinking which benefited me a lot.

References
[1] Wei Wei, Zhou Laishui, Zhang Liyan. Rapid STL File Loading and Visualization. Mechanical Science and Technology, 2006, 8: 935-938
[2] WANG Zeng-bo. Research of 3D Model Visualization Based on STL Files. Journal of Hengyang Normal University, 2008, 6: 21-24
[3] Liu Jinyi, Hou Baoming. Efficient Topological Reconstruction of Solids in STL Format. Journal of engineering graphics, 2003, 4: 34-39
[4] Quigley N, Lyne J E. Development of a Three-Dimensional Printed, Liquid-Cooled Nozzle for a Hybrid Rocket Motor. Journal of Propulsion & Power, 2015, 6: 1-2
[5] Campbell T A, Ivanova O S. 3D Printing of Multifunctional Nanocomposites. Nano Today, 2013, 2: 119-120
[6] Giannatsis J, Dedoussis V. Additive Fabrication Technologies Applied to Medicine and Health Care: A Review. Int J Adv Manuf Tech, 2009, 1-2: 116-127
[7] PAN Hai-peng ZHOU Tian-rui HU Shi-fei et al. Implementation of the visualization for STL model. Journal of Plasticity Engineering, 2006, 6: 94-98
[8] WANG Congjun ZHANG Lichao HUANG Shuhui. A New STL Data Compressed Storage Format. china mechanical engineering, 2001, 5: 558-560
[9] LI Jiang-feng , ZHONG Yue-xian, LI Dian-sheng. Research on errors identifying and repairing of STL file. Machinery Design & Manufacture, 2002, 2: 40-42
[10] Hong Taoqing. A Discussion on the Uniform Equation of the Cylindrical Section, Journal of Lishui University, 2008, 05: 72-74
[11] XIAO Renxian ZHANG Junjian FENG Hao et al. STL Files Reading, Displaying and Manipulating. Journal of Ceramics, 2011, 02: 285-288
[12] DING Dian-kun, BIAN Ping-yong. The Parameter Equation of Space Curve's Projection on the Plane. Journal of Luoyang Teachers College, 2005, 5: 23-24
[13] Li Busong, Zhou Gang, Wang Congjun et al. Realization of arithmetic for dividing solid model based on STL file format. Journal of Huazhong University of Science and Technology(Nature Science), 2002, 3: 40-42
[14] Xie Cunxi, Li Zhongyang, Cheng Xiaoyang. Research on the Establishment of adjacency relation of STL file and its slicing algorithm. Journal of South China University of Technology, 2000, 3: 33-38
[15] Cline A K, Renka R J. A Storage-Efficient Method for Construction of a Thiessen Triangulation. University of Texas at Austin, 1982. 119-140
[16] Fang T P, Piegl L A. Algorithm for Delaunay Triangulation and Convex-Hull Computation Using a Sparse Matrix. Comput Aided Design, 1992, 8: 425-436
[17] Zhou Xiaoyun, He Dazeng. The Algorithm of Triangulation of Scattered Data Points on the Plane. Journal of Computer-Aided Design and Computer Graphics, 1994, 4: 256-259
[18] Zhao Dongmei, Zhang Jialei. Complex multiplication and rotation matrix. Journal of Chifeng Unimersity, 2011, 6: 22-23