A New Slicing Procedure for Rapid Prototyping Systems

Y.-S. Liao[1] and Y.-Y. Chiu[2]

[1]Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan; and [2]Department of Industrial Design, Oriental Institute of Technology, Taipei, Taiwan

The current slicing rule of most rapid prototyping (RP) systems cannot ensure unilateral tolerances on the whole prototype and often results in problems of overcut and undercut on the same part. This drawback leads to unsatisfactory precision of the part in post processing. In order to reduce the above problems, a new slicing method is proposed in this paper. Based on the geometry information in a stereolithography (STL) file, an algorithm is developed. The appropriate slicing rule is selected according to the inner product of the normal vector and working direction of the part, together with the function of the part to be manufactured. The STL file is cut into 2D sections and an accurate contour is calculated. After the slicing computation, an appropriate working path is produced. The rules proposed in this paper have been verified. This work contributes to the improvement of slicing rules in existing RP systems, especially in systems using uniform thickness slicing. It also improves manufacturing efficiency and working tolerances.

1. Introduction

Based on the principle of layer manufacturing, the rapid prototyping (RP) technique starts with the intersection of a 3D model from CAD (usually a stereolithography (STL) file) with layers of 2D horizontal planes. As a result, a stack of 2D geometry contours is obtained, each representing a cross-section of the 3D model. Next, a computer takes the bottom slice of the geometric contour and transmits different levels of laser energy to be raw materials on the working plane, after having placed the raw material on the location as required by the geometric contour. The raw material is filled on one slice after another from the bottom up, and the process is repeated until a complete 3D part is produced. Theoretically, given the very

Correspondence and offprint requests to: Y.-S. Liao, Department of Mechanical Engineering, National Taiwan University, 1 Roosevelt Road, Sec. 4, Taipei, 10617 Taiwan. E-mail: liaoys@ccms.ntu.edu.tw

small thickness of each layer of the 2D slice, the 3D part should be identical with the 3D model constructed by the computer. However, in reality, materials used in RP are all limited to a minimum thickness, thus generating staircase error. There are also other factors that may affect RP precision. Since these errors have been discussed in great detail in previous studies, they are not repeated here [1,2].

The part produced by RP presents a stepwise shape, which is more evident as the thickness of the 2D slices increases. Precision becomes worse when the contour is gentle, and machining precision becomes better as the contour becomes steeper. Variation of the shape of a part is thus an important factor for precision. In addition, the presence of fine features on the workpiece which are smaller than the thickness of a layer also affects the machining precision. Current studies on the improvement of precision are discussed in the following sections.

Dolenc and Makela [3] and Kulkarni and Dutta [4] used the original CAD file in studies of slicing methods. The former searched for the optimal slicing thickness for a variety of workpiece features. The latter derived the optimal slicing thickness based on the curvature of the workpiece and the cusp height formed between the layers. Sabourin et al. [5,6] divided a workpiece into an "outer region" in contact with the boundary, and an "inner region" without contact with the boundary. The "outer region" affects precision. Thus, a variety of slicing thicknesses are used based on the variation of workpiece contour shape. A thinner slice is used for a workpiece of a flatter contour slope. In contrast, a thicker slice is used when the workpiece contour slope becomes steeper. Since the "inner region" does not affect precision, thicker slices are used than for the "outer region" to increase machining efficiency.

Hope et al. [7] presented a TruSurf system for a five axis numerical control (NC) waterjet cutting machine for cutting the layers. An inclined surface is obtained by controlling the orientation of the waterjet cutter replacing the staircase shape and reducing the size of the error. Hope et al. [8] used this system and took the midpoint of a layer in a curved workpiece as the basis for calculating the machining contour and deciding the slope of the cutting vector, but it still could not ensure unilateral tolerance on the whole prototype.

The approaches adopted in previous studies all focus on the premiss of minimising the tolerance. They all overlook the importance of tolerance uniformity. This often results in the problems of overcut and undercut on the same workpiece. That is, the tolerance cannot be ensured in the finished workpiece, and this leads to difficulty in post-processing. For post-processing, which is essential, a proper working tolerance is necessary for RP. If all the tolerance is left outside the original workpiece contour, it is called "positive tolerance", and when all the tolerance is left inside the original workpiece contour, it is called a "negative tolerance". In other words, positive tolerance leaves extra material, and thus is also called "oversize," as shown in Fig. 1(*a*). Negative tolerance leads to insufficient material, and thus is called "undersize," as shown in Fig. 1(*b*). In practice, both types of tolerance have specific users. For most RP applications such as industrial mock-ups and investment casting, positive tolerance is necessary because the materials left can be removed in post-processing to meet precision requirements. However, for certain industrial applications such as cores used in producing a cavity, no positive tolerance is required. Instead, overcut is required to make the finished product smaller than the actual dimensions, so that the duplicated cavity has extra tolerance for post-processing. Hence, negative tolerance is required in this case. These important factors, which have previously been ignored, must be considered for RP improvement.

For most RP techniques, which are limited by the consideration of working efficiency or process technologies, uniform slicing is the most commonly adopted basic setting to meet the requirement of working efficiency. A slice thickness of around 0.1 mm is commonly used at present. Take an RP process which uses the gluing of sheets (e.g. LOM by Helisys, or SAHP by KIRA) as an example. Since sheets of uniform thickness are used as the material for machining, material thickness cannot be changed according to variations in the features of a part. Hence, this approach loses fine features and further affects the dimensional accuracy. Other powder or liquid RP systems also use materials of uniform thickness, in most cases to increase working efficiency, and hence the same problems are produced.

Based on the above review, it is essential to achieve unilateral tolerance on the whole prototype to ensure precision and to retain fine features. In this paper, a new slicing rule for uniform thickness is developed to ensure that the entire workpiece has either a positive or negative tolerance. The method proposed in this paper can be applied in a variety of RP systems, especially when uniform slice thickness is used.
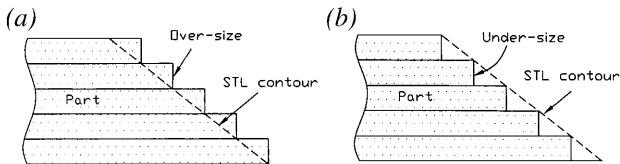

Fig. 1. Illustration of cases of (*a*) oversize and (*b*) undersize.

2. Slicing Strategies

The current RP technique adopts two strategies when using the slicing method to produce a 3D part. The most common approach is top down slicing. In this method, the intersection point between the triangle facet and the slicing plane is calculated and extended down to the thickness of the layered material. These steps are repeated to form a 3D part. The other approach is bottom up slicing, which extends from the intersection point up to the thickness of the layered material. This can be considered as the extension up from the intersection point between the slicing plane of the previous layer and triangle facet to the thickness of the layered material. Repeating these steps can also form a 3D part. Calculations of the intersection point for both methods are described in the next section.

2.1 Calculation Methods for Top Down and Bottom Up Slicing

At present, the STL file has become the standard format for RP, including ASCII and binary forms [9,10]. The STL file describes the normal vector of the triangle facet and the coordinates of the three vertices of a triangle facet. The normal vector refers to the direction on the outside the part or of the waste material. Refer to Fig. 2, where the intersection point between the slicing plane and the triangle facet is drawn. The STL ASCII format is given as follows:

solid name	[description]
facet normal $n_i n_j n_k$	name: the STL filename
outer loop	$n_i n_j n_k$: the unit normal vector of facet
vertex $v1_x v1_y v1_z$	$v1_x v1_y v1_z$: the coordinates of vertex 1
vertex $v2_x v2_y v2_z$	$v2_x v2_y v2_z$: the coordinates of vertex 2
vertex $v3_x v3_y v3_z$	$v3_x v3_y v3_z$: the coordinates of vertex 3
endloop	
endfacet	
endsolid name	

As shown in Fig. 2, let L be the length between two points of coordinates $(v1_x, v1_y, v1_z)$ and $(v2_x, v2_y, v2_z)$. The slicing plane $(Z = Z_p)$ passes through L, and (X_p, Y_p, Z_p) are the intersection points between the slicing plane and L. Then, by parametric representation, the following relation is obtained:

$$\frac{X_p - v1_x}{v2_x - v1_x} = \frac{Y_p - v1_y}{v2_y - v1_y} = \frac{Z_p - v1_z}{v2_z - v1_z} = t \quad (0 \le t \le 1) \quad (1)$$

By manipulated Eq. (1), equations for top down slicing and bottom up slicing can be derived, and are given in Eqs (2) and (3), respectively.

Top down slicing [11]

$$X_p = \frac{(Z_p - v1_z) \times (v2_x - v1_x)}{(v2_z - v1_z)} + v1_x$$

$$Y_p = \frac{(Z_p - v1_z) \times (v2_y - v1_y)}{(v2_z - v1_z)} + v1_y \quad (2)$$
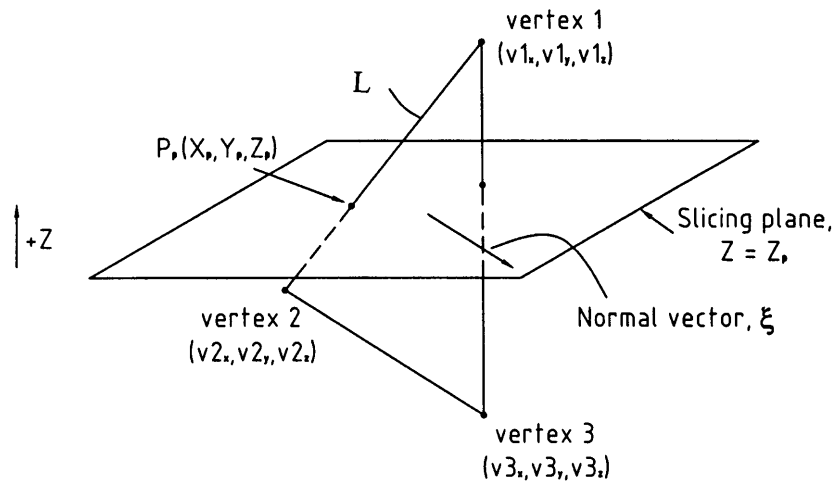
Fig. 2. The intersection point between the slicing plane and the triangle facet.

Z_p = slicing height

Bottom up slicing

Z_p = slicing height − one layer thickness

= slicing height of the previous layer

if $Z_p \geq v2_z$

$$X_p = \frac{(Z_p - v1_z) \times (v2_x - v1_x)}{(v2_z - v1_z)} + v1_x$$

$$Y_p = \frac{(Z_p - v1_z) \times (v2_y - v1_y)}{(v2_z - v1_z)} + v1_y \qquad (3)$$

if $Z_p < v2_z$

$$X_p = \frac{(Z_p - v2_z) \times (v3_x - v2_x)}{(v3_z - v2_z)} + v2_x$$

$$Y_p = \frac{(Z_p - v2_z) \times (v3_y - v2_y)}{(v3_z - v2_z)} + v2_y$$

Figure 3 shows the elevational cross-sectional view of the interface between the part and the waste material. In the figure, there are two different regions. The area enclosed by the dashed line is the cross-section of a 3D part. The dotted area is the cross-section of the part after the slicing of the layers. As shown in Figs 3(a) and 3(b), owing to slope variations, both the top down slicing and bottom up slicing methods, result in conditions that are out of control, namely "undersize" and "oversize," respectively. It will be impossible to obtain an accurate shape and satisfactory results by sanding the resulting part. In other words, the popular top down slicing method can create "undersize" or "oversize" problems depending on the slopes given. With a negative slope, vertical cutting by a laser along the intersection point results in undersize. On the other hand, with a positive slope, vertical cutting by laser along the intersection point results in oversize. Only for a part with a 90° slope can the exact size be obtained. Undersize makes the machined size smaller than the expected value, whereas oversize leaves extra material after machining. As discussed earlier, both methods have their own unique applications. From the
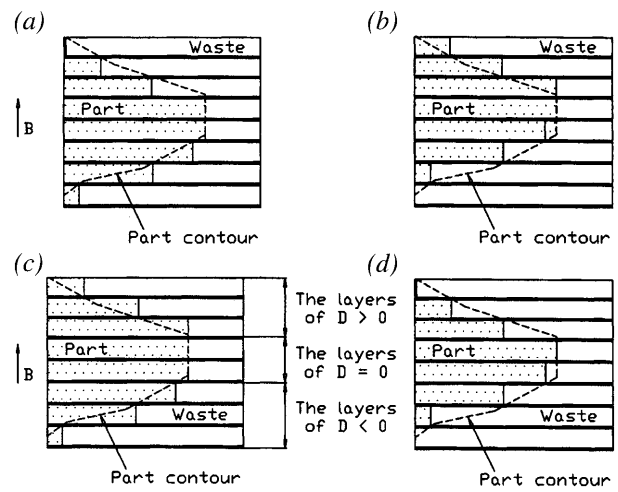

Fig. 3. The elevational cross-sectional view of the interface between the part and the waste material. This is a case of RP of gluing sheets. (a) Top down slicing. (b) Bottom up slicing. (c) Positive tolerance slicing. (d) Negative tolerance slicing.

above discussion, an attempt to place all the tolerance on the same side is essential for practical use.

2.2 Slicing Strategies for Positive Tolerance and Negative Tolerance

Generally speaking, the geometric information of a 3D part cannot be described fully by the slope of the part alone. Decisions made based only on the part's slope may lead to faulty results. When the part and the waste material are exchanged, as shown in Figs 3(a) and 3(b), the expected over-size turns into an undersize, leading to results opposite to the original expectation. This is an obvious example. Feygin and Pak [12] developed a slicing rule based on the positive or negative value of the slope of the part to ensure the presence of an extra working tolerance for post processing. However, this often results in the above-mentioned ambiguous conditions. Thus, a method based on the slope of the part is not suitable.

To solve these problems, a positive tolerance slicing method and a negative tolerance slicing method are proposed. The results of slicing are shown in Figs 3(c) and 3(d), respectively.

As mentioned earlier, an STL file is capable of describing the normal vectors and the coordinates of the three vertices of a trianglar facet. Referring to Fig. 4, let the unit vector of the RP working direction be **B**, the normal vector of each triangle face be ξ, and the angle between each triangle face and horizontal axis be θ. The user determines the preferred orientation based on such factors as machining precision, machining time, part strength, and difficulty of waste material removal (for LOM). Next, the orientation of each triangle face is calculated based on the inner product (dot) value of the normal direction of each triangle face and the RP working direction (**B**), i.e. Eq. (4) shows the results derived from the generally used top down slicing method.

$$\xi \bullet B = D \tag{4}$$

where, $\xi = n_i\mathbf{i} + n_j\mathbf{j} + n_k\mathbf{k}$, and **B** is the working direction of the RP process.

Then,

if $D = 0$, the intersection point at z value is the exact contour.

if $D < 0$, the intersection point at z value is the positive tolerance.

if $D > 0$, the intersection point at z value is the negative tolerance.

Hence, if **D** = 0 during the slicing process, this means that the contour is vertical. In other words, the contour of the intersection point overlaps the original contour and so it maintains the exact contour without producing any error. If **D** < 0 (a positive slope in the case of Fig. 4), this implies that the angle between the contour and the working direction is smaller than 90°, and cutting along the intersection point contour naturally forms a positive tolerance. If **D** > 0 (a negative slope in the case of Fig. 4), this implies that the angle between the contour and working direction is larger than 90°, and cutting along the intersection point contour causes a negative tolerance.

Hence, if the user requires a positive tolerance and finds **D** > 0 in the calculation, the original intersection point should be replaced by the intersection point of the previous layer (z = slicing height − one layer thickness) (i.e. the bottom up slicing method mentioned earlier). If **D** < 0, top down slicing should be used to ensure the extra working tolerance and avoid
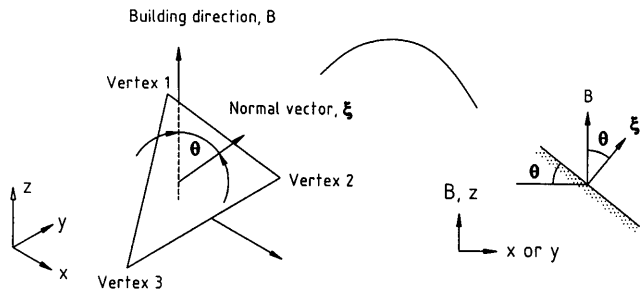

Fig. 4. Schematic diagram of triangle facet and the working direction, ξ is the normal vector of facet, B is the working direction.
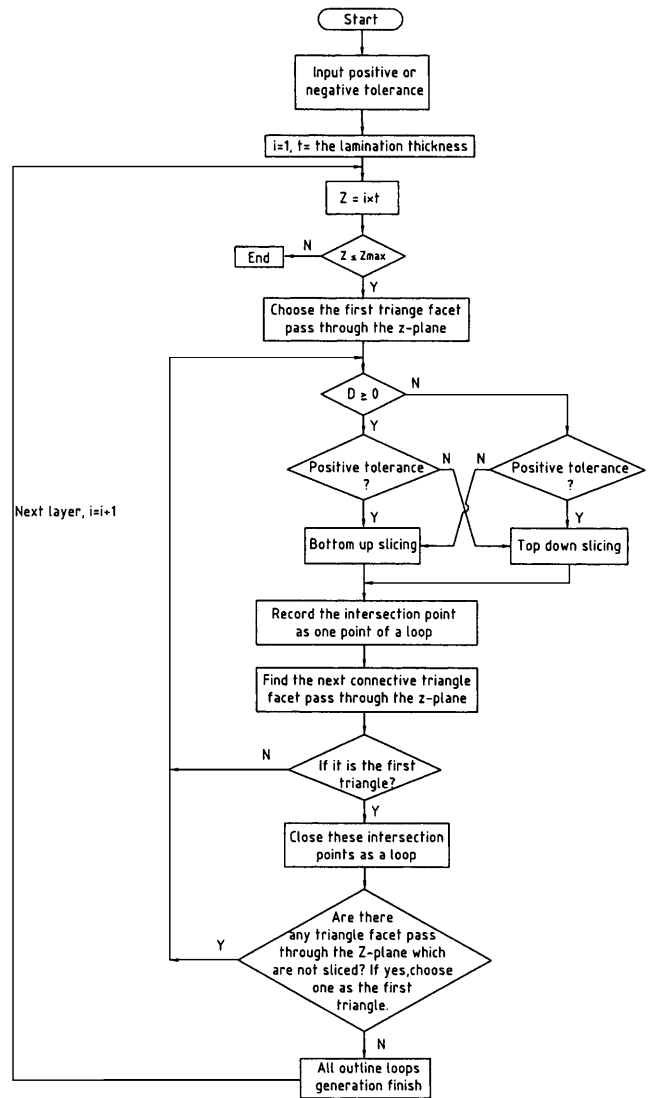

Fig. 5. Flowchart of the slicing rule.

the problem of overcut. When **D** = 0, this implies a correct contour, and either the top down or bottom up slicing method results in the same intersection point. The method is expressed in Eq. (5). Similarly, when a user requires a negative tolerance and finds **D** < 0, then the bottom up slicing method should be used. If **D** > 0, then use the top down slicing method to ensure overcut. Again, when **D** = 0, this implies a correct contour, then either the top down or bottom up slicing method results in the same intersection point. The method is expressed in Eq. (6).

Positive tolerance

if ($D \geq 0$)

calculating intersection point by

"bottom up slicing method"

else   (5)

calculating intersection point by

"top down slicing method"

Negative tolerance

> if ($D \geq 0$)
>
> > calculating intersection point by
> >
> > "top down slicing method"
>
> else   (6)
>
> > calculating intersection point by
> >
> > "Bottom up slicing method"

3. Slicing Procedures

The above concept can be explained by the flowchart shown in Fig. 5.

First of all, input an STL file. Then choose a suitable working tolerance depending on the workpiece function. If the workpiece is to be used as an industrial mock-up, positive tolerance is required. If it is to be used as a core, a negative tolerance is selected. Then, one layer is added at a time, based on the lamination thickness. Starting from the first layer thickness, take any one triangle facet that passes through the z-plane as the first triangle facet for calculating the intersection point. Next, determine the dot value (**D**) between the normal direction of the triangle facet and the working direction. Given the selection of positive tolerance, use bottom up slicing if **D** $\geq$ 0, and top down slicing if **D** < 0. In the case of negative tolerance, use top down slicing if **D** $\geq$ 0, and bottom up slicing if **D** < 0. The intersection point derived is then recorded as one point of the loop. Next, search for the connective triangle facet that passes through the z-plane through the common edge between triangle facets. Determine whether the triangle facet is the first triangle facet already sliced. If not, use the aforementioned procedures to locate another intersection point. If yes, connect these intersection points to form a closed loop. Next, search for any triangle facet that passes through the z-plane but has not been sliced yet. If there are any, this means that there is another loop. In this case, select one of the triangle facets as the first triangle facet and find other intersection points using the same method to form a closed loop. If there are none, this indicates that all triangle facets that pass through the z-plane have been sliced. Finally, these loops form a 2D cross-section, which is the slicing intersection curve that we are looking for to serve as the working path. Then, add another layer of lamination thickness and calculate the slicing

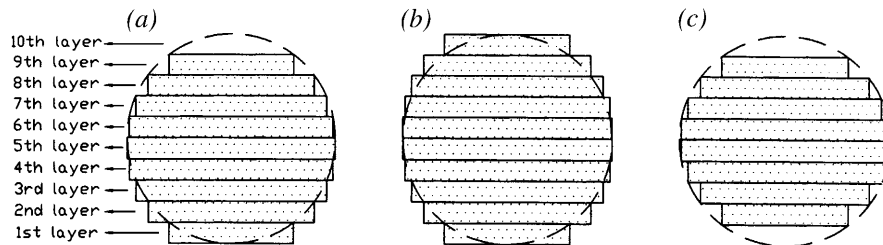**Table 1.** Comparison among different slicing methods.

Layer	Dot value (**D**)	Slicing methods: top down (T) or bottom up (B)		
		Ordinary slicing method	Positive tolerance	Negative tolerance
1	−0.9	T	T	B
2	−0.7	T	T	B
3	−0.5	T	T	B
4	−0.3	T	T	B
5	−0.1	T	T	B
6	0.1	T	B	T
7	0.3	T	B	T
8	0.5	T	B	T
9	0.7	T	B	T
10	0.9	T	B	T

intersection curve for the next layer. Repeat these procedures till the maximum z-axis value of the workpiece is reached.

4. Experimental Verification

To prove that the above flowchart is correct, an STL file with a circular cross-section is presented for verification. The circular cross section has a diameter of 2 mm, a slicing thickness of 0.2 mm, and a total of 10 layers of slicing thickness. The upward vertical shown in Fig. 6 denotes the unit vector of the working direction, i.e. the **B** value in Eq. (4).

The slicing method developed in this study is compared with the top down slicing method commonly used, as shown in Fig. 6. In the figure, the dashed line inside is the cross-section of the part to be made, whereas the dotted area denotes the cross-section of the part after slicing. Table 1 lists the dot values and the slicing methods for positive and negative tolerances. It is obvious that the top down slicing method cannot ensure the correct location of the working path. Some parts of it are outside the workpiece contour, while others are inside the contour, as shown in Fig. 6(*a*) and Table 1. If the stair shape is eliminated during post-processing, the circular cross-section turns into an oval shape, resulting in a deviation in shape. The slicing method for positive tolerance presented in this study uses the slicing approach based on different dot values. In this case, top down slicing is used for layers 1 to 5 with a negative dot value; bottom up slicing is then used

Fig. 6. The comparison between different slicing methods. (*a*) Top down, (*b*) positive tolerance, and (*c*) negative tolerance slicing.
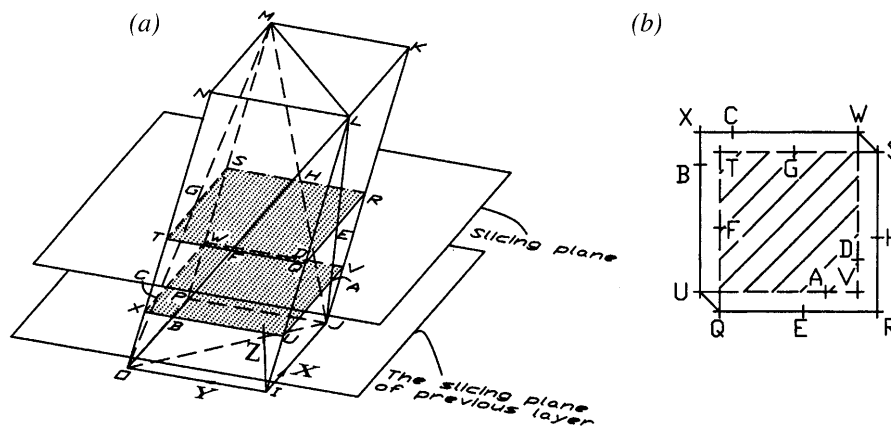
Fig. 7. (a) The schematic diagram of the inclined cuboid and the slicing plane. (b) The shaded area denotes the cross-section of the part after negative tolerance slicing. The solid line inside is the cross-section of the part after positive tolerance slicing.

for layers 6 to 10 with a positive dot value. This approach ensures that the working path is completely outside the work-piece contour to leave it oversize for post-processing, as shown in Fig. 6(b). In the slicing method for negative tolerance, since the dot value is negative between the 1 and 5 layers, bottom up slicing is used; the dot value is positive between the 6 and 10 layers, so top down slicing is used. This ensures that the working path is completely inside the workpiece contour giving undercut, as shown in Fig. 6(c).

Take another example of a 3D part for method verification. An STL file with an inclined cuboid and slicing planes is shown in Fig. 7(a). The inclined cuboid is formed by 12 triangle facets. Two dotted areas denote the cross-section of the part after slicing. The dotted area enclosed by the *QFTGSHREQ* region is calculated by the top down slicing method, and the dotted area enclosed by the *UBXCWDVAU* region is calculated by the bottom up slicing method. Figure 7(b) shows that the 2D cross-section is obtained by the intersection between the cuboid and the slicing plane. The shaded area denotes the cross-section calculated by the negative tolerance slicing method, and the area enclosed by the *UBXCWSHREQ* region is the cross-section calculated by the positive tolerance slicing method. Table 2 gives the geometric information of the inclined cuboid STL file and the dot value

(**D**). Table 3 gives the intersection points of coordinates after slicing.

The calculation of positive tolerance slicing is described as follows. First, choose one triangle facet *OIL* that passes through the slicing plane. The positive value of **D** (**D** = 0.2425) is obtained after calculating the inner product of the normal vector of the triangle facet and the working direction. Because of the specified positive tolerance slicing, the intersection point *U* is calculated by the bottom up slicing method. Then, choose the connective triangle facet *OLN* that passes through the slicing plane through the common edge between triangle facets, and calculate the intersection point *B* by the bottom up slicing method based on the positive value of **D** (**D** = 0.2425). Similarly, choose the triangle facets *MON* and *MPO*, then the intersection points, *X*, *C*, and *W* are obtained by the bottom up slicing method based on the positive value of **D** (**D** = 0.2425). Similarly, choose the triangle facet *MJP*, then the intersection point *S* is obtained by the top down slicing method based on the negative value of **D** (**D** = −0.2425). Then, choose the triangle facet *KJM*, and calculate the intersection point *H* by the top down slicing method based on the negative value of **D** (**D** = −0.2425). Repeat similar procedures, choose the triangle facets *JKL* and *IJL*, then the intersection points, *R*, *E*, and *Q* are obtained by the top down slicing method based on the

Table 2. The geometric information of an inclined cuboid STL file and inner product value (*D*). The working direction (*B*) is <0, 0, 1>.

Triangle facet	The geometric information of an STL		Dot value (**D**)
	Normal vector	The corresponding three vertices of the coordinates	
OIL	<−0.9701,0,0.2425>	(−7.36,7.36,−3.86), (−7.36,−2.64,−3.86), (−2.64,−7.36,15)	0.2425
OLN	<−0.9701,0,0.2425>	(−7.36,7.36,−3.86),(−2.64,−7.36,15),(−2.64,2.64,15)	0.2425
IJL	<0,−0.9701,−0.2425>	(−7.36,−2.64,−3.86), (2.64,−2.64,−3.86), (−2.64,−7.36,15)	−0.2425
JKL	<0,−0.9701,−0.2425>	(2.64,−2.64,−3.86), (7.36,−7.36,15), (−2.64,−7.36,15)	−0.2425
KJM	<0.9701,0,−0.2425>	(7.36,−7.36,15), (2.64,−2.64,−3.86), (7.36,2.64,15)	−0.2425
MJP	<0.9701,0,−0.2425>	(7.36,2.64,15), (2.64,−2.64,−3.86) (2.64,7.36,−3.86)	−0.2425
MPO	<0,0,0.9701,0.2425>	(7.36,2.64,15), (2.64,7.36,−3.86)	0.2425
MON	<0,0,0.9701,0.2425>	(7.36,2.64,15), (−7.36,7.36,−3.86), (−2.64,2.64,15)	0.2425

Table 3. The intersection points of coordinates after slicing calculation.

Point	Coordinates	Point	Coordinates	Point	Coordinates
T	(−5.14,5.14)	S	(4.86,5.14)	V	(3.61,−3.61)
F	(−5.14,0.45)	G	(−0.45,5.14)	D	(3.61,−1.56)
Q	(−5.14,−4.86)	X	(−6.39,6.39)	W	(3.61,6.39)
E	(0.16,−4.86)	B	(−6.39,4.35)	C	(−4.35,6.39)
R	(4.86,−4.86)	U	(−6.39,−3.61)		
H	(4.86,−0.16)	A	(1.56,−3.61)		

negative value of $\mathbf{D}$ ($\mathbf{D} = -0.2425$). Finally, the region enclosed by the points *UBXCWSHREQ*, as shown in Fig. 7(*b*), is the cross-section of positive tolerance slicing.

The negative tolerance slicing plane shown by the region enclosed by points *QFTGSWDVAU*, as shown in Fig. 7(*b*) can be obtained similarly.

5. Conclusion

A new slicing method based on uniform thickness, which adopts the inner product of the normal vector of an STL file and the working direction as the decision rule is proposed in this paper. Depending on the workpiece functional requirement for either a positive or a negative tolerance, the top down or bottom up slicing method is selected to ensure unilateral tolerance for the whole surface of the prototype. Experimental results have shown that there are two major effects when using this method. First, it improves the disadvantages of undercut and overcut which occur in the various RP slicing methods, and also the precision quality of the completed workpiece can be ensured after the post-processing procedures that are essential to RP. Secondly, slicing was conducted with uniform thickness to achieve the dimension precision requirements and to increase the working efficiency significantly.

References

1. W. Cheng, F. Y. H. Fuh, A. Y. C. Nee, Y. S. Wong, H. T. Loh and T. Miyazawa, "Multi-objective optimization of part-building orientation in stereolithography", Rapid Prototyping Journal, 1(4), pp. 12–23, 1995.
2. G. M. Fadel and C. Kirschman, "Accuracy issues in CAD to RP translations", Rapid Prototyping Journal, 2(2), pp. 4–17, 1996.
3. A. Dolenc and I. Makela, "Slicing procedures for layered manufacturing techniques", Computer-Aided Design, 26(2), pp. 119–126, 1994.
4. P. Kulkarni and D. Dutta, "An accurate slicing procedure for layered manufacturing", Computer-Aided Design, 28(9), pp. 683–697, 1996.
5. E. Sabourin, S. A. Houser and F. H. Bohn, "Adaptive slicing using stepwise uniform refinement", Rapid Prototyping Journal, 2(4), pp. 20–26, 1996.
6. E. Sabourin, S. A. Houser and F. H. Bohn, " Accurate exterior, fast interior layered manufacturing", Rapid Prototyping Journal, 3(2), pp. 44–52, 1997.
7. R. L. Hope, P. A. Facobs and R.N. Roth, "Rapid prototyping with sloping surfaces", Rapid Prototyping Journal, 3(1), pp. 12–19, 1997.
8. R. L. Hope, R. N. Roth and P. A. Facobs, "Adaptive slicing with sloping layer surfaces", Rapid Prototyping Journal, 3(3), pp. 89–98, 1997.
9. StereoLithography Interface Specification, 3D Systems, Valencia, CA, 1988.
10. M. Burns, Automated Fabrication – Improving Productivity in Manufacturing, chap. 6, pp. 227–231, Prentice Hall, 1993.
11. R. C. Luo, W. Z. Lee, P. T. Yu, H. T. Leong and Y. F. Lin, "3D CAD cross-sectional slicing approaches for rapid prototyping systems", The Symposium of 11th National Conference on Automation Technology, pp. 1153–1160, National Chung Cheng University, Chia-Yi, Taiwan, July 1999.
12. M. Feygin and S. S. Pak, "Laminated object manufacturing apparatus and method", United States Patent 5 876 550, 1999.