Locally Adjusting Operation on Concave Vertexes of Planar Polygon

Huabing Wang¹, a, Weijun Liu²,b and Hongyou Bian²,c

¹Mechanical School, Huaihai Institute of Technology, Lianyungang, Jiangsu Province, RP China
²Advanced Equipment Research Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, Liaoning Province, RP China

¹w_huabing@163.com, bwjliu@sia.cn, c bhy@sia.cn

Keywords: Planar Polygon, Offset Operation, Concave Vertexes.

Abstract. Triangular mesh is one of the widely used digital representation modes in CAD and CAM. And the offset operation of planar polygon takes an important role in path planning of layer milling complicated surfaces represented by triangular mesh. A vertex classification based adjusting method is presented. It improves the performance of edge isometry based offset algorithm.

Introduction

Calculating the offset contours of planar polygons is one of the most important step in tool path planning of layer milling and data process of rapid prototyping. There are many researches and achievements in the offsetting algorithms of planar polygon. The implemented algorithms can be divided into such three types as edge isometry based, Voronoi Graph based and offset line search based.

Edge isometry based offset algorithm is very intuitive, easy to understand, obvious geometrically with its calculation process and strong adaptively. But there is also some shortcoming for it. Its efficiency is not high, because it must deal with the complex operation to connect the offset edges, to remove ther redundant rings and to conduct a large amount of validity tests for avoidance of interference. Hansen and Arbab[1], Suh and Lee [2] simplified the judgement of the interferent segments and improved the reliability and efficiency of this algorithm. Zhang [3, 4] implemented such algorithm in the rapid prototyping data process of laser spot radius compensation.

Yan[5] and Kim[6] used the Voronoi Graph to calculate the offset contours of planar polygons. In most cases of implementing this algorithm there is no need of the judgement and removal of self-intersecting rings. But the unavoidable calculation of angular bisector results in heavy time-consuming work. And the complicated judgement and computation is to be involved in the application of this kind of algorithm for the complex concave polygons or the boundaries of complex connected region[4].

The basic idea of offset line search based algorithm is to guarantee the least distance between any point on the offset contours and any point on the original polygons equal to given offset distance. It is a relatively successful algorithm. But the necessary calculation of distance makes its speed low.

The biggest question for the Voronoi Graph based offset algorithm is that it is time consuming because the computation of the Voronoi Graph for planar polygons is complicated and the further operation of intersecting points solving and offset edges searching is rather complex. In the application of offset line search based algorithm there are many times of calculation of the distance between the point and the polygons which is a heavy burden for computer. So the edge isometry based offset algorithm is chosen as the study object. Zhao used this type of algorithm to complete the path planning of stereo lithography[7] and Zhang applied it in path planning of metal powder rapid prototyping[3].
After triangular mesh of the work piece is layerly sliced by a set of planes, a group of regular planar polygons are received. These polygons can be divided into inner rings and outer rings. The counter-clockwise is assigned to the outer rings as the positive direction and the clockwise for the inner rings. There are at least one outer ring and a number of inner rings in one layer contour. In the data processing of some traditional rapid prototyping systems such as SLA there exists polygon contour offset operation. But the offset distance is usually fairly short, so the offset contours are near to the original ones and they resemble each other. And after the operation of simple offsetting and linking, the resulted offset contour can fulfill the demand of the system. Even if there were some self-intersecting rings, the condition could be very simple.

To accomplish the roughing tool path planning of layer milling with flat end cutter, it is necessary to twice make the offset operation to the slice contours in order to generate the loop cutting tool center path and the contours for the calculation of end points of the scanning lines. The offset distance here is more longer than that in traditional prototyping. So there may be a lot of locally and globally self-intersecting rings in the process of offsetting. If there were no proper process before offset opertion and did it as in the traditional prototyping, the resulted offset contours would be very confusing and be hard to pick up the valid rings.

A polygon composed of only one outer ring is offset outerward with different distances, and the resulted contour polygons are illustrated in Fig. 1. The edge isometry based offset algorithm is adopted, in which the edges offset from the ones connected by convex vertex are linked with line and the concave vertexes are moved on their corresponding bisectors. Where the offset distance is short enough such as in fig.1(a), the offset contour is near and similar to the original one and only simple self-intersection occurs. With the offset distance becoming longer as in fig.1(b), (c), (d) the self-intersection phenomena become more complex and it is almost impossible to find out some proper rules to resolve the offset contours and to eliminate the invalid portions.

So a vertex locally adjusting method is proposed to improve the edge isometry based offset algorithm. The flow chart of the new algorithm is illustrated in fig.2.

**Polygon vertex locally adjusting method**

According to the offset distance and the type of vertex determined by the relative position of its neighboring vertexes, the locally adjusting operation of the vertex will be done. For the polygons in a slicing layer, the vertex locally adjusting operation flow chart is given in fig.3.

The vertexes can be classified into three kinds, which are convex vertex, flat vertex and concave vertex illustrated in fig.4. A discriminant \( \Delta = \vec{v}_1 \times \vec{v}_2 \) for the the classification of vertexes is designed, where \( \vec{v}_1 \) is the unit vector of directional edge \( v_{i-1} v_i \) and \( \vec{v}_2 \) is the unit vector of directional edge \( v_{i-1} v_i \).
A small real number as $10^{-5}$ is assigned as the threshold value. If $\Delta > 10^{-5}$, the vertex $v_i$ is convex and no operation is to be made. If $|\Delta| \leq 10^{-5}$, the vertex $v_i$ is flat and it is to be eliminated from the vertex array. If $\Delta \leq -10^{-5}$, the vertex $v_i$ is concave and it is to be processed further in detail.

![Diagram of Polygon Offset Flow](image1)

![Diagram of Layer Polygon Vertex Adjusting](image2)

![Diagram of Convex, Flat, and Concave Vertices](image3)

![Diagram of Critical Edge Length Calculation](image4)

What to do with the concave vertex is determined by the relative position of its neighbor vertices and the offset distance $r$. A critical edge length is calculated according to the offset distance $r$ and the position vectors of $v_i$, $v_{i-1}$ and $v_{i+1}$.

As illustrated in fig.5(a), draw a $r$ radius circle in the included angle area of the neighbor edges of vertex $v_i$, let its center on the bisector of the angle and let it tangent to these two edges or their extended line respectively on point $v_{i1}$ and point $v_{i2}$. Now the length between $v_i$ and $v_{i1}$ or $v_i$ and $v_{i2}$ is defined as the critical edge length $L_c$. 
Sign the length of the edge $v_{i-1}v_i$ with $L_1$ and the length of the edge $v_iv_{i+1}$ with $L_2$. Compare $L_c$ with $L_1$ and $L_2$. And according to the comparing results the concave vertexes can be classified into four kinds as shown in fig.6.

The first kind of concave vertex is given in fig.6(a), in which the lengths of the two neighbor edges of the concave vertex $v_i$ are both greater than the critical edge length $L_c$. And no adjustment for this kind of vertex is to be done.

The second kind of concave vertex is illustrated in fig.6(b), in which $L_1$ is smaller than $L_c$ and $L_2$ is not. The position of this kind of concave vertex $v_i$ need to be changed. The vertex moving principle is that the offset contours should resemble the original polygons as possible and this adjustment should not cause undercut phenomenon. The vertex moving process can be divided into two steps as following.

In the first step, the vertex is moved along the direction of $\vec{v}_2$ by a certain length to $v_i$. Then recalculate the critical edge length $L_c$ for $v_i$ and the forward neighbor edge length $L_1$. Let $L_c$ equal to $L_1$, as shown in fig.7(a).

In the second step, the unit vector of the ahead neighbor edge $v_{i-2}v_{i-1}$ of the vertex $v_{i-1}$ is labeled as $\vec{v}_0$. The next processing is determined by the classification according to the relative position of the edges of $v_{i-2}v_{i-1}$, $v_{i-1}v_i$ and $v_iv_{i+1}$.

If $\vec{v}_0 \times \vec{v}_1 > 10^{-5}$, then the relative position belongs to the type shown in fig.7(b). The adjusting operation ends for vertex $v_i$.

If $|\vec{v}_0 \times \vec{v}_1| \leq 10^{-5}$ and $\vec{v}_0 \times \vec{v}_2 < 0$, then the relative position looks like the type shown in fig.7(c), in which the vertexes $v_{i-2}$, $v_{i-1}$ and $v_i$ are colinear. The vertex $v_{i-1}$ is eliminated from the vertex array and in the next cycle the No. i vertex is to be processed.

If $\vec{v}_0 \times \vec{v}_1 < -10^{-5}$ and $\vec{v}_0 \times \vec{v}_2 < -10^{-5}$, then the relative position can be classified as the type illustrated in fig.7(d). The vertex $v_i$ is to be moved to the point $v_j$, the intersecting point of the extended line of $v_{i-2}v_{i-1}$ and the reverse extended line of $v_{i}v_{i+1}$, then the vertex $v_{i-1}$ is deleted and the No. i-1 and No. i vertexes are readjusted.

If $\vec{v}_0 \times \vec{v}_2 \geq -10^{-5}$ and $\vec{v}_0 \times \vec{v}_3 < -10^{-5}$, in which $\vec{v}_3$ is the unit vector of $v_{i-1}v_{i+1}$, the relative position belongs to the type shown in fig.7(e). Delete the vertex $v_i$ from the vertex array and readjust the No. i-1 and No. i vertexes.

**Fig.6. Classification of concave vertex**

(a) $L_1 \geq L_c$, $L_2 \geq L_c$
(b) $L_1 < L_c$, $L_2 \geq L_c$
(c) $L_1 \geq L_c$, $L_2 < L_c$
(d) $L_1 < L_c$, $L_2 < L_c$

**Fig.7. Adjustment of No.2 type of concave vertex**

(a) (b) (c) (d) (e) (f)
If $\vec{v}_0 \times \vec{v}_2 \geq -10^{-5}$ and $\vec{v}_0 \times \vec{v}_1 \geq -10^{-5}$, then the relative position can be classified as the type illustrated in fig.7(f). The vertex is to be removed from the vertex array and the No. i-2, No. i-1 and No. i vertexes are to be readjusted.

The 3rd kind of concave vertex is illustrated in fig.6(c), in which $L_1$ is no smaller than $L_c$ and $L_2$ is. The position of this kind of concave vertex $v_i$ need to be changed in two steps as following.

In the first step, sign the unit vector of the directional edge $v_{i+1}v_i$ with $\vec{v}_1$ and that of $v_{i+1}v_{i+2}$ with $\vec{v}_2$. Move the vertex $v_i$ along the reverse direction of $\vec{v}_1$ to $v_i$ and let the critical edge length at $v_i$ is equal to the edge length of $L_2$ as illustrated in fig.8(a).

In the second step, sign the unit vector of the directional edge $v_{i+1}v_i$ with $\vec{v}_3$. What to do next is attribute to the relative position of the neighbor edges belong to the type illustrated in fig.8(c), in which the vertexes $v_i, v_{i+1}$ and $v_{i+2}$ are colinear. The vertex $v_{i+1}$ is eliminated from the vertex array and the adjusting operation for vertex $v_i$ completes.

If $\vec{v}_2 \times \vec{v}_3 < -10^{-5}$ and $\vec{v}_1 \times \vec{v}_3 < -10^{-5}$, the relative position belongs to the type shown in fig.8(d), in which the edge $v_{i+1}v_{i+2}$ takes up some position in the circle that is given in fig.8(a). If the cutter working on the point of $v_i$, it will cut in the contour represented by the edge of $v_{i+1}v_{i+2}$. So further processing for the vertex $v_i$ is necessary. The vertex $v_i$ is moved to the crossing of the extended line of edge $v_{i+1}v_i$ and the reverse extended line of edge $v_{i+1}v_{i+2}$ that is labeled $v_i$, and then the vertex $v_{i+1}$ is deleted from the vertex array. Then the No. i vertex is readjusted.

If $\vec{v}_2 \times \vec{v}_3 \geq -10^{-5}$ and $\vec{v}_2 \times \vec{v}_3 > -10^{-5}$, in which $\vec{v}_4$ is the unit vector of from vertex $v_{i+1}$ to vertex $v_{i+2}$, the relative position looks like the type shown in fig.8(e). Eliminate the vertex $v_i$ from the vertex array and readjust the No.i and No.i-1 vertexes.

If $\vec{v}_2 \times \vec{v}_3 < -10^{-5}$ and $\vec{v}_2 \times \vec{v}_3 > -10^{-5}$, the relative position of the neighbor edges belong to the type illustrated in fig.8(f). The vertex $v_{i+1}$ is to be eliminated from the vertex array, and then the No.i and No. i+1 vertexes are to be readjusted.

![Fig.8. Adjustment of No.3 type of concave vertex](image)

The 4th kind of concave vertex is illustrated in fig.6(d), in which $L_1$ and $L_2$ are both smaller than $L_c$. The position of this kind of concave vertex $v_i$ need to be changed in four steps as following.

Step 1. Compare $L_1$ and $L_2$. If $L_1 > L_2$, turn to step 3, else if $L_1 = L_2$, turn to step 4, else go on with step 2.

Step 2. Here $L_1 < L_2$, as is shown in fig.9, in which the circles have the same radius of $r$ equal to the offset distance and the circle centered $o_1$ is tangent to the lines of edge $v_{i+1}v_i$ and edge $v_{i+1}v_{i+2}$. And $v_{i+2}$ is the point of tangency of the circle and the line including $v_{i+1}v_i$. Then parallel to $v_{i+1}v_i$ move the circle center to $o_2$, and let the vertex $v_{i+1}$ lie on the circle, sign the new point of tangency with $v_{i+2}$, label $L_c = |v_{i+2}|$. If $L_2 > L_c$, then adjust the vertex $v_i$ by the way as the same as that of the second type of concave vertex, as is illustrated in fig.9(a). If $L_2 \leq L_c$, as is shown in fig.9(b) and fig.9(c), turn to step 4.
Step 3. Here $L_1 > L_2$, as is shown in fig.9, in which the circles have the same radius of $r$ that is equal to the offset distance and the circle centered $o_1$ is tangent to the lines of edge $v_i-1v_i$ and edge $v_iv_{i+1}$. And $v_{t1}$ is the point of tangency of the circle and the line including $v_i-1v_i$. Then parallel to $v_i-1v_i$ move the circle center to $o_2$, and let the vertex $v_{i+1}$ lie on the circle, sign the new point of tangency with $v_{t2}$, label $L_c = |v_{t1}v_{t2}|$. If $L_1 > L_c$, then adjust the vertex $v_i$ by the way as the same as that of the third type of concave vertex, as is illustrated in fig.10(a). If $L_1 \leq L_c$, as is shown in fig.9(b) and fig.9(c), turn to step 4.

Step 4. Eliminate the vertex $v_i$ from vertex array. And go on with the adjusting operation for the No.1-1 and No. i vertices.

Thus classify the vertexes of the polygons according to the relative positions to their neighbor vertexes and the given offset distance. Different adjusting strategies are adopted for different kinds of vertexes as shown above and at the same time it is necessary to take recursive adjusting operation for the neighbor vertexes influenced by the vertex adjusting.

**Conclusion**

The concave vertexes with relatively acute angles have been eliminated or substituted after the vertexes were locally adjusted. So it is avoided that the vertex reflects in the offset process. Comparing with the method that the concave vertexes are all deleted, the proposed one can give more accurate offsetting contours. The proposed method is more stable than the one presented in [7]. It has been used successfully in the milling tool path planning software for a subtractive rapid prototyping system.

**Acknowledgements**

This work was financially supported by the Young Teacher Innovation Foundation of Huaihai Institute of Technology (KQ10117).

**References**

[1] A. Hansen, F. Arbab: ACM Transactions on Graphics Vol. 12 (1992), p. 152.
[2] Y. S. Suh, K. Lee: Computer-Aided Design Vol. 22 (5)1990, p. 273.
[3] Z. Jiayi, Doctoral Dissertation of Shenyang Institute of Automation, Chinese Academy of Sciences (2005), in Chinese.
[4] Z. Jiayi, H. Yongping and L. Weijun: Journal of Engineering Graphics Vol. 27 (2006), p. 16, in Chinese.
[5] Y. Bing, L. Bibo, D. Zhiyun, Z. Dawei and C. Ziping: Journal of CAD & CG Vol 11 (1999), p 66, in Chinese.
[6] D. S. Kim: Computer-Aided Design Vol. 30 (1998), p 1047.
[7] Z. Jibin, Doctoral Dissertation of Shenyang Institute of Automation, Chinese Academy of Sciences (2004), in Chinese.