

A Method to Generate Exact Contour Files for Solid Freeform Fabrication

Sashidhar Guduri, Graduate Research Assistant
Richard H. Crawford, Assistant Professor
Joseph J. Beaman, Professor
Dept. of Mechanical Engineering
University of Texas at Austin

Abstract

Existing methods to create contour files generate a polygonal approximation of the contours instead of an exact representation. This paper presents a method to generate exact contours from Constructive Solid Geometry (CSG) representations. The method preserves the accuracy of the contour files provided the primitives used to generate the CSG tree are polygonal or quadric objects. Due to the inclusion of quadric objects into the primitive set an additional effort to solve for the intersection points between two quadric curves is required. The paper also presents a method to convert piecewise quadratic contours to toggle point files for raster scanning solid freeform fabrication processes.

Introduction

Computer software plays a major role in Solid Freeform Fabrication (SFF) techniques. The main aim of the computer software is to generate contour files at different cross sections of a given part geometry. These contour files can be rasterized to generate laser toggle point files.

Present methods for generating contour files use faceted representations of the part geometry. The contour files generated by these methods are polygonal approximations of the actual contours. This paper presents a method to generate exact contours, instead of polygonal approximations, by using a higher order geometric description of the part.

Description of the Method

Part Geometry Representation

The form of the geometric description of mechanical parts to be produced by SFF significantly affects the accuracy of the final part. The current state-of-the-art for most SFF technologies, consists of tessellating the surfaces of the geometric model into a mesh of non-overlapping triangular facets. The resulting geometry is transmitted in a standard file format, the so-called STL file format, established by 3D Systems, Inc. [1]. This format has been adopted by many CAD vendors, is readily available, and is considered adequate for most visualization applications. However, for producing accurate patterns and functional parts, the adequacy of the STL format is unclear [3]. There is a trade-off between the accuracy and the size of the geometric description. Highly non-linear surfaces, such as those found on turbine blades, manifolds, etc., must be tessellated into a large number of small facets, resulting in very large data files, and the accuracy of such descriptions is still suspect. For this reason richer geometric descriptions must be introduced to the SFF community.

Research at The University of Texas is aimed at using higher order geometric descriptions as input to the Selective Laser Sintering (SLS) process. Higher order geometric descriptions will increase the accuracy while reducing the file size. The research reported herein focuses on processing Constructive Solid Geometry (CSG), which represents the part geometry a binary tree whose leaf nodes are primitive solids (e.g. spheres, cylinders, cones etc.) and whose non-terminal nodes are the Boolean set operations union (\cup), intersection (\cap), and difference ($-$). In this work, processing for SFF proceeds in two stages: (1) slicing the input geometry to obtain the part's boundary curves in each scanning plane and (2) scan converting each contour to produce laser toggle points for raster scanning the part. Each of these stages is described below.

Slicing

The SLS process produces parts on a layer-by-layer basis. The part geometry is sliced by sintering planes (planes perpendicular to z axis) at regular intervals, typically 0.005-0.01 inches apart. Slicing the part geometry, represented by a CSG tree, at different cross sections involves slicing the primitives individually, and generating a slice for each primitive. The slice of a primitive is a collection of curves of degree two or less. Figure 1 shows the slices obtained by slicing a sphere and a cylinder having different orientations.

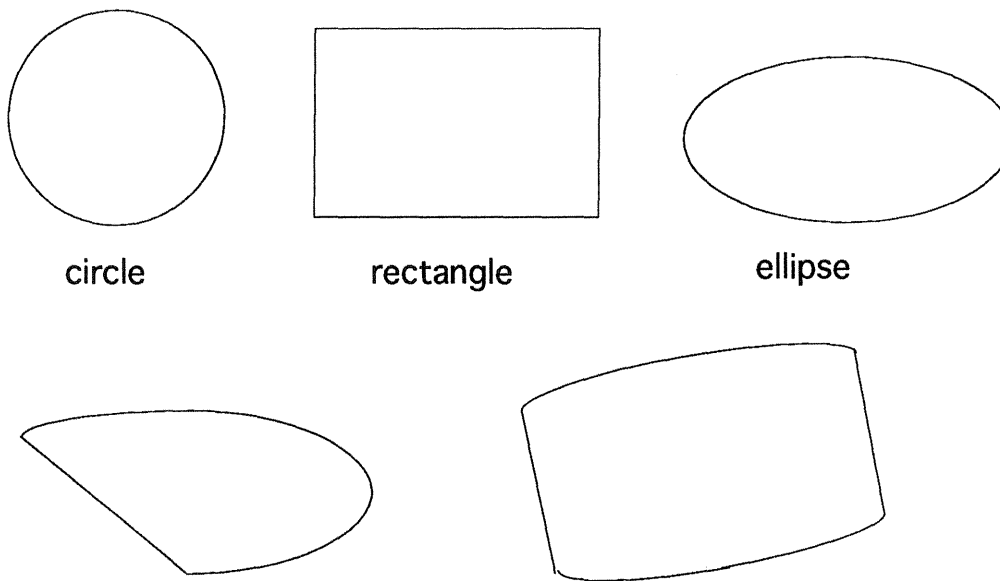


Figure 1. Slices obtained by slicing a sphere and a cylinder

Boolean operations

Once the slices for all the primitives are obtained, Boolean set operations are performed on them to generate the contour for that cross section. The process of applying the Boolean operations to simplify the binary tree is recursive. First, at the bottom level of the binary tree, the slices are generated by intersecting the current sintering plane with the primitives at the leaf nodes, producing contours for the primitives. Then the next higher level contours are generated by applying Boolean operations to the first level contours. This process is repeated recursively level by level to generate the final contour

for the current cross section. The process of applying a given Boolean operation to two given contours requires the calculation of the intersection points of the contours. An outline of the method to calculate the intersection points is given in Appendix A. After the calculation of all the intersection points, each contour is decomposed into curve segments whose end points are the intersection points. The curve segments of one contour are then classified with respect to the other contour, and vice-versa. The process of classification is described in the next section.

Classification

After the calculation of the intersection points, the curve segments must be grouped appropriately according to the Boolean operation (union, intersection or subtraction). This is done by classifying each curve segment on one contour with respect to the other contour. Classification is the process of determining if a particular curve segment is inside or outside the other slice. A ray casting algorithm is used for this classification [4]. According to this algorithm a point chosen on the curve segment is as the base point and an in-plane ray is generated. If the number of intersections of the ray with the other contour is even, then the point is outside the contour; otherwise, the point is inside. The special cases of tangency and vertex intersections are taken care by the software. After classification, the curve segments that form the new contour boundary are identified, based on the particular Boolean operation:

- Operation \cup : Curve segments of both the contours that are classified as out.
- Operation \cap : Curve segments of both the contours that are classified as in.
- Operation $-$: Curve segments that are classified as out for the first contour and in for the second contour.

Figure 2 illustrates the steps in the classification and set operation application.

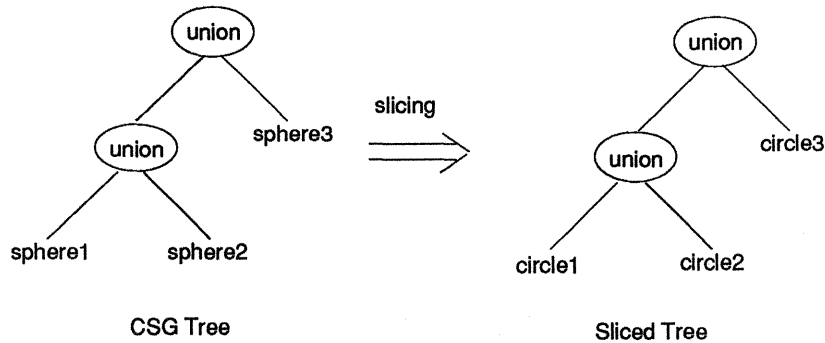
Toggle Point Generation

The process of toggle point generation for each layer is similar to rasterization in computer graphics image generation. The process generates discrete points on the contours at which the energy beam must be toggled on or off. The process consists of intersecting mathematical rays directed along each scan line in the contour plane with each curve segment of the contour and requires solving a second degree equation in one unknown. These intersections are then ordered in increasing order of the scan line coordinate direction. Special cases such as vertex intersection and ray tangency with the curve are taken care by the software. The output of the process is a binary file of toggle points.

Conclusions

Software for processing geometric data is an important factor in the development of Solid Freeform Fabrication techniques. This paper describes a method to generate exact contour files from CSG geometry for the SLS process. Future work focuses on expanding the variety of sources for geometric data, including development of a contour generation algorithm for rational bicubic parametric surfaces.

sphere1 : center at (0, 0, 0) and unit radius
 sphere2 : center at (1, 0, 0) and unit radius
 sphere3 : center at (0, 1, 0) and unit radius



sliced at z = 0.0

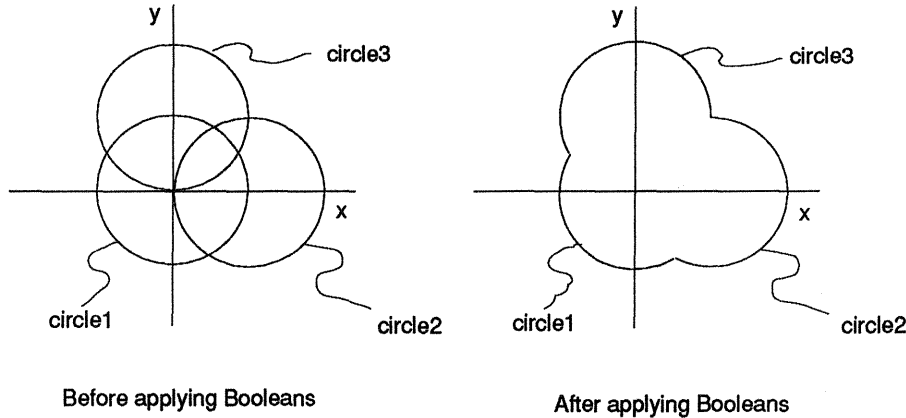


Figure 2. Contour classification and set operations.

Appendix A

Equation of a quadric curve in a plane can be written as

$$I(x,y) = ax^2 + by^2 + cxy + dx + ey + f = 0 \quad (\text{eq. 1})$$

Parametrization

The non-trivial case of parametrizing equation 1 is when a and b are non-zero. Otherwise the equation is already linear in one variable and expressible as a rational polynomial expression of the other, and hence a rational parametrization. To obtain the rational parametrization one has to make equation 1 non-regular in x or y . That is, eliminate the quadratic term in either x or y through coordinate transformation. Then one variable is linear and expressible as a rational polynomial expression of the other. The steps shown below describe an algorithm to eliminate the y^2 term.

- (1) If equation 1 has a real root at infinity, a linear transformation of the type

$$x = a_1x + b_1y + c_1 \text{ and } y = a_2x + b_2y + c_2$$

will suffice. If equation 1 has no real root at infinity, one must use a fractional linear transformation of the type

$$x = (a_1x + b_1y + c_1) / (a_3x + b_3y + c_3) \text{ and}$$

$$y = (a_2x + b_2y + c_2) / (a_3x + b_3y + c_3)$$

- (2) Points at infinity for equation 1 correspond to linear factors of the degree form (highest degree terms) of equation 1. For a quadric curve this corresponds to a real root at infinity if $c^2 > 4ab$. Otherwise both roots at infinity are complex. Furthermore, $c^2 = 4ab$ corresponds to a polynomial parametrization for the quadric curve, as the degree form is a perfect square.
- (3) Applying a linear transformation for $c^2 \geq 4ab$ gives rise to

$$I(x, y) = I(a_1x + b_1y + c_1, a_2x + b_2y + c_2).$$

To eliminate the y^2 term, choose b_1 and b_2 such that

$$ab_1^2 + cb_1b_2 + bb_2^2 = 0.$$

Here both the values of b_1 and b_2 can always be chosen to be real.

- (4) After applying a homogeneous linear transformation for $c^2 < 4ab$, to eliminate the y^2 term, choose b_1 , b_2 and b_3 such that they satisfy the homogeneous conic:

$$ab_1^2 + bb_2^2 + cb_1b_2 + db_1b_3 + eb_2b_3 + fb_3^2 = 0.$$

The values of b_1 and b_2 are both real if $(cd - 2ae)$ is not less than the geometric mean of $4af - d^2$ and $4ab - c^2$.

- (5) Finally, choose the remaining coefficients a_i s and c_i s, ensuring that the appropriate transformation is well defined.

The equations for two quadric curves are of the form

$$a_1x^2 + b_1y^2 + c_1xy + d_1x + e_1y + f_1 = 0$$

$$a_2x^2 + b_2y^2 + c_2xy + d_2x + e_2y + f_2 = 0$$

One of these equations can be parametrized using the method given above. For a detailed method refer to [2]. The parametric equations of x and y can be substituted in the second equation to give a polynomial of fourth degree in a single variable. The real roots of the polynomial can be obtained using the method described below.

Solutions of a fourth degree polynomial in one variable

A fourth degree polynomial can be written in the form

$$c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 = 0$$

When $c_4 = c_3 = 0$, the equation can be solved using the familiar quadratic formula. For the case when $c_4 = 0$, the equation reduces to a cubic that can be solved as follows:

- (1) Divide the equation by c_3 to get the normal form

$$x^3 + Ax^2 + Bx + C = 0$$

- (2) Substituting $x = y - \frac{A}{3}$ eliminates the quadratic term, yielding

$$y^3 + 3py + 2q = 0.$$

- (3) Using Cardano's formula, the determinant is

$$D = q^2 + p^3 \text{ and } u, v = \sqrt[3]{-q \pm \sqrt{D}}$$

the roots of which are

$$y_1 = u + v \text{ and } y_{2,3} = -\frac{u+v}{2} \pm \frac{\sqrt{3}}{2}(u-v)i$$

Solving a quartic equation

Dividing the general equation of a quartic by c_4 yields the normal form:

$$x^4 + Ax^3 + Bx^2 + Cx + D = 0$$

Substituting $x = y - \frac{A}{4}$ eliminates the cubic term, resulting in:

$$y^4 + py^2 + qy + r = 0.$$

The resolvent cubic is then

$$z^3 - (p/2)z^2 - rz + (rp/2) - (q^2/8) = 0.$$

With z being the root of this equation, the roots of the quartic can be obtained by solving the two quadratic equations

$$y^2 \pm y\sqrt{2z-p} + z \mp \sqrt{z^2-r} = 0$$

Resubstituting yields the correct values of x . For a detailed description of this method, refer to [5].

References

1. 3D Systems, Inc., "Stereolithography Interface Specification", 3D Systems, Inc., Valencia, CA, June 1988.
2. Abhyankar, S. S. and Bajaj, C., "Automatic Rational Parametrization of Curves and Surfaces I: Conics and Conicoids", Technical Report, Purdue University.
3. Darrah, J. and Wielgus, M., "A New CAD Model Format for SFF Machines?", *Proceedings of Solid Freeform Fabrication Symposium*, 1990, J. J. Beaman, H. L. Marcus, D. L. Bourell, and J. W. Barlow, eds., Austin, TX, August 6-8, pp. 121-125.
4. Foley J. D., van Dam, A., Feiner, S. K., and Huges, J. F., *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, 2nd edition, 1990, pp 92-99.
5. Glassner, A. S, *Graphics Gems*, Academic Press, Boston, 1990, pp 404-406.