

# Non-uniform Offsetting and its Applications in Laser Path Planning of Stereolithography Machine

Yong Chen

Epstein Department of Industrial and Systems Engineering  
University of Southern California, Los Angeles, CA 90089

Reviewed, accepted August 21, 2007

## ABSTRACT

Laser path planning is an important step in solid freeform fabrication processes such as Stereolithography (SLA). An important consideration in the laser path planning is to compensate the shape of laser beam. Currently the compensation is divided into two steps, Z-compensation and X-Y compensation, and the shape of laser beam is assumed to be uniform for the whole platform. In this research, we present a sampling based non-uniform offsetting method which accounts for the different shapes of laser beam at various locations. We discuss the related steps and algorithms. We demonstrate its effectiveness by using various test cases. Besides improving the accuracy of SLA machine, non-uniform offsetting can also be applied to address other accuracy issues caused by thermal and structural variations.

## 1. INTRODUCTION AND MOTIVATION

It is well known that the accuracy of a Stereolithography machine varies from the center to the borders of a platform. To achieve better accuracy in various applications, the users are suggested to put their parts as close to the center of the platform as possible. For example, although the platform of a *Viper HR* machine ([www.3dsystems.com](http://www.3dsystems.com)) is 10.25 x 10.25 inch in *X* and *Y* direction, the platform area which can build part with high accuracy is about 5 x 5 inch. These areas are called the “*sweet spots*” in the platform.

There are two major reasons for the existing of “*sweet spots*” in the SLA machines.

- (1) The distance from the mirror to the resin surface varies from the center to the border ( $L$  verse  $L'$  as shown in Figure 1). This may affect laser's focus;
- (2) The laser enters the resin surface in an angle other than vertical to the surface ( $\alpha$  as shown in Figure 1). This may change the cured resin shape related to the laser beam.

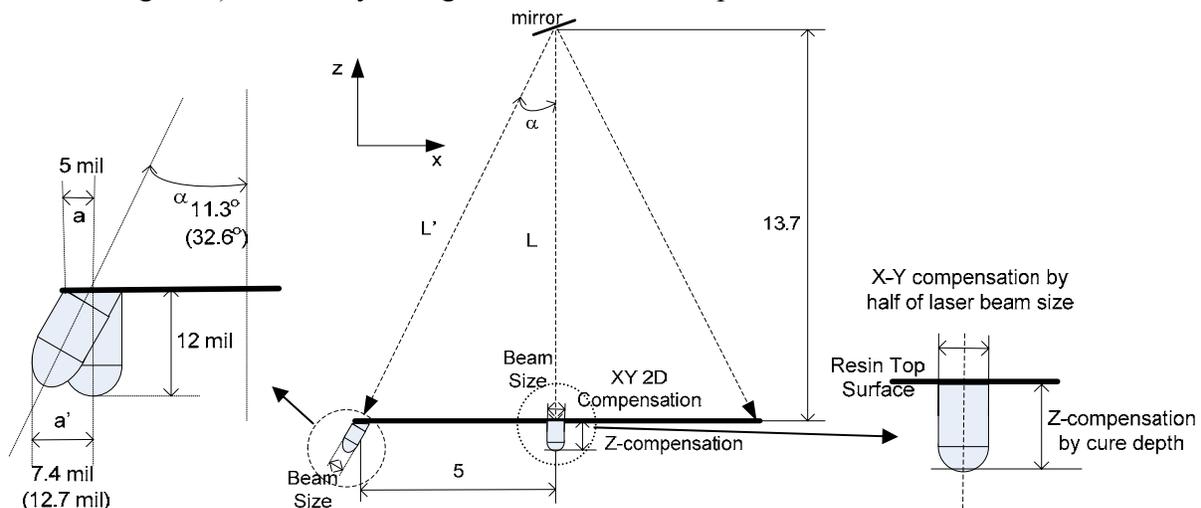


Figure 1: Illustration of laser beams in the platform of a Stereolithography machine.

In the newly developed *Viper-Pro* machine ([www.3dsystems.com](http://www.3dsystems.com)), laser focus can be dynamically adjusted. This addresses the first problem, that is, the laser may be out of focus in the border areas of the platform. However, the second problem, changing cured resin shape due to different entering angle of a laser beam, has never been addressed in tool path planning before. Currently, *3D Systems*' slicing algorithm assumes all the laser beams have a uniform bullet shape as shown in Figure 1 (right). Therefore, the tool path compensation due to the shape of a laser beam can be divided into two independent steps [Jacobs 1992][Jacobs 1996]:

(1) X-Y compensation by half of a laser beam's size: for example, for a laser beam's size of 0.01'', the sliced 2D contours are offset by 0.005''. This is a 2D offsetting problem. The offset contours are the laser' drawing path for the current layer;

(2) Z-compensation by a given cure depth: for example, for a cure depth of 0.016'' and layer thickness of 0.004'', the slicing contours of four consecutive layers are used in a set of Boolean operations. The calculated Boolean results, which compensate the cure depth of a laser beam, are the final drawing path of a laser beam.

However, systematic accuracy errors exist by treating laser beams as a uniform bullet shape instead of varying by different entering angles. As shown in Figure 1 (left), if we assume the cured resin shape as a rotated uniform bullet by an entering angle  $\alpha$ , the accuracy error can be significant. For example, for a *Viper HR* machine with a platform size 10 x 10 inch, the maximum error is  $\sim 0.0024''$  corresponding to a maximum  $\alpha$  around  $11.3^\circ$ ; while for a *Viper Pro* machine with a platform size 30 x 25 inch, the maximum error can be  $\sim 0.0077''$  corresponding to a maximum  $\alpha$  around  $32.6^\circ$ .

In this paper, we present a non-uniform offsetting approach to address the changing entering angle of a laser beam at different locations. We address the tool path calculation in order to compensate the different laser beam shapes.

The central contributions of this paper are:

(1) we systematically define a non-uniform offsetting problem, which is related to but different from problems such as uniform offsetting, general *Minkowski* operation, volume sweeping, and dynamic implicit surfaces;

(2) we propose a general framework for the non-uniform offsetting problem;

(3) we presented a sampling point based method and related algorithms to calculate the non-uniform offsetting boundary.

For an input STL file, the proposed method will generate a valid offset model as the tool path for the input model. Therefore it is easy to integrate our method into the current usage of SLA systems. By considering different laser beam shapes at various locations, the accuracy difference related to the platform positions (i.e., within or outside "*sweet-spots*") will be significantly smaller. This is especially important for the *Viper-Pro* system, which has a comparably big platform size. The research is also a step toward the goal of the same accuracy within the whole platform. This is an important requirement for all Rapid Manufacturing applications.

## 2. PROBLEM FORMULATION

As shown in Figure 2 (left), when  $\alpha=0$ , it is an appropriate simplification by treating a laser beam shape as a set of 2D circles for multiple layers. Consequently, the 3D tool path planning can be divided into two compensation processes, X-Y compensation and Z-compensation. However, when  $\alpha$  is comparably big, the simplification is not appropriate any more. As shown in Figure 2 (right), the compensation values at different slicing layers are slightly different. For

example, the laser beam size at layer 1 is the same or close to the case of  $\alpha=0$ ; while there is a noticeable error at layers 3 and 4.

In addition, compensation values and orientations are different depending on its positions in the platform. As shown in Figure 2 (right), angle  $\alpha$  varies at different locations. Consequently, the laser beam shapes constantly change. Some examples of projected laser beam shapes on  $XY$  plane are shown in Figure 2 (right). It's obvious that the laser beam orientations and sizes to be compensated are quite different at various locations.

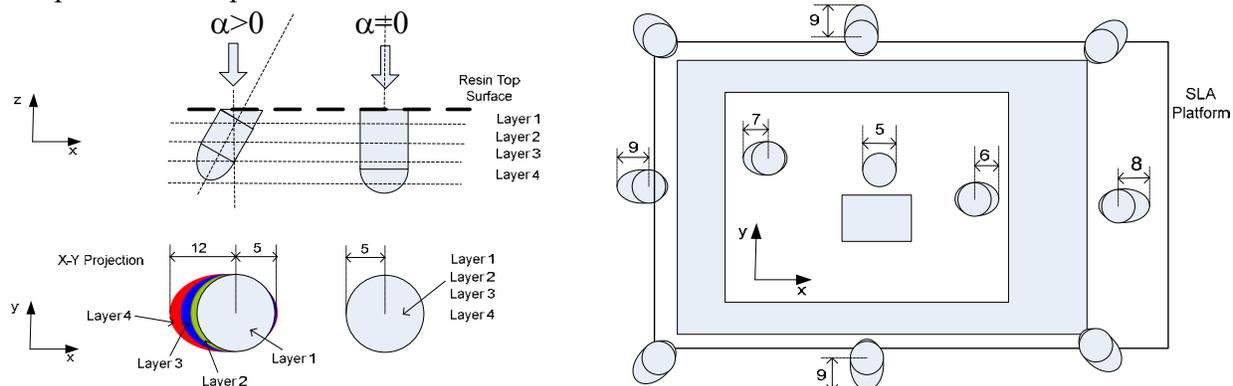


Figure 2: Illustration of laser beams in the platform of a Stereolithography machine.

In this paper, we use a general non-uniform offsetting problem to formulate the above laser shape compensation problem. Offsetting operations are special cases of *Minkowski* sums and differences. Suppose two sets  $A$  and  $B$  are closed and regular subset of Euclidean space  $E^2$  or  $E^3$ . The Minkowski sum of  $A$  and  $B$ , denoted  $A \oplus B$ , is defined as  $C = A \oplus B = \{a + b \mid a \in A, b \in B\}$ , where “+” denotes the normal vector addition of two points. The *Minkowski* difference, denoted  $A \otimes B$  is  $\overline{A \oplus B}$ . Usually both  $A$  and  $B$  are given with fixed geometries [Farouki *et al.* 2000]. A challenge in our problem is that only one shape ( $A$ ) is fixed while another shape ( $B$ ) has different geometry depending on its relative position within the platform. Suppose the laser drawing path is  $h(x, y, z)$ . For a point  $v(x, y, z)$  on  $h(x, y, z)$ , the corresponding laser shape is defined as  $g(x, y, z)$ . The laser paths and shapes will form a geometry  $f(x, y, z)$  which can be defined as:

$$f(x, y, z) = h(x, y, z) \oplus g(x, y, z).$$

However, in our tool path planning problem, we need to calculate the laser drawing path  $h(x, y, z)$  for a given geometry  $f(x, y, z)$  to be fabricated. According to the property of *Minkowski* operations [Ghosh 1993], we can define  $h(x, y, z)$  as:

$$h(x, y, z) = f(x, y, z) \oplus (g(x, y, z))^{-1},$$

where  $(g(x, y, z))^{-1}$  is the reciprocal set of  $g(x, y, z)$ .

Let  $\mathbf{o}$  denote the origin point of the coordinate system in which a convex polytope  $B$  is placed. Clearly  $B \oplus \{\mathbf{o}\}$  is equal to  $B$ , since set  $\{\mathbf{o}\}$  is a singleton point set. Therefore,  $B \oplus B^{-1} = \{\mathbf{o}\}$ . That is,  $(g(x, y, z))^{-1}$  is actually the symmetrical set of  $g(x, y, z)$  with respect to the origin point. For every point  $v \in g(x, y, z)$ , there exist a point  $v' \in (g(x, y, z))^{-1}$  such that  $v + v' = \mathbf{0}$ , where “+” denotes vector addition of two points, and  $\mathbf{0}$  denotes zero vector which is equivalent to the origin point. In addition, if we use “sense” of a normal to specify whether the normal is diverging outward from the object or converging inward, the object  $B^{-1}$  will have the negative sense. The above operations are illustrated in Figure 3 with two different shapes of  $g(x,$

$y, z$ ). Notice  $g(x, y, z)$  and related  $(g(x, y, z))^{-1}$  are symmetrical around origin points  $\mathbf{o}_1$  and  $\mathbf{o}_2$  respectively.

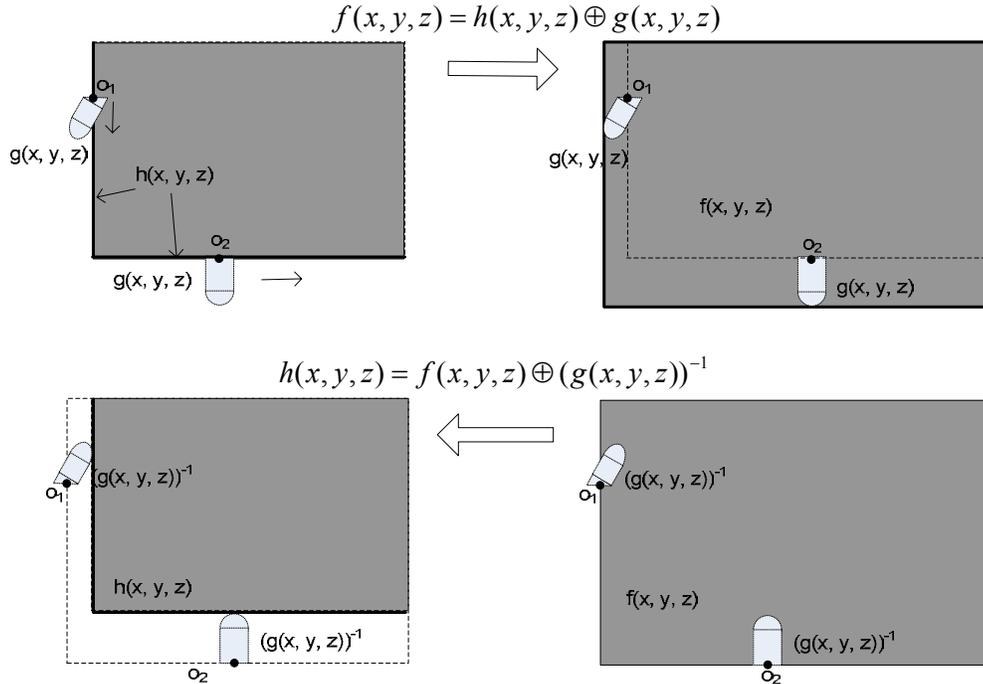


Figure 3: Illustration of reciprocal set for *Minkowski* operations.

The computational process for *Minkowski* operations may also be facilitated by transforming the equations in the following forms:

$$A \oplus B = \bigcup_{b \in B} A_b, \text{ where “}\cup\text{” denotes set union operation.}$$

$$A \otimes B = \bigcap_{-b \in B} A_{-b}, \text{ where “}\cap\text{” denotes set intersection operation.}$$

The non-uniform offsetting is a quite difficult and unique problem. (1) It is different from general offsetting problem studied by [Rossignac and Requicha 1986] [Chen *et al.* 2006], in which a uniform offsetting distance  $r$  is given. Therefore, we can use a ball with radius  $r$  to represent  $g(x, y, z)$  for all  $v(x, y, z)$ . Since the shape of  $g(x, y, z)$  is fixed as a sphere, the problem can be simplified based on some geometric properties related to the ball. (2) Most literatures related to general *Minkowski* operations also only studied the cases in which both objects  $A$  and  $B$  are fixed. (3) The non-uniform offsetting problem has similarity to the volume sweeping problem which can be defined as:  $\text{Sweep}(A \text{ on } B) = \bigcup_{b \in B} A @ M(b)$  [Kim *et al.* 2003], that is, swept volume is the volume generated by sweeping a solid ( $A$ ) along a smooth trajectory ( $B$ ). Even though some researcher studied changing the shapes of sweeping body along the trajectory, the volume sweeping problem is generally simpler since the trajectory  $B$  is usually a 3D curve. (4) The non-uniform offsetting problem is also different the dynamic implicit surfaces studied in level set methods [Osher and Fedkiw, 2003]. For the dynamic implicit surfaces, each point along the boundary can constantly change its moving speed and direction. Therefore, a dynamic programming with multiple iterations is generally required. In comparison, in our problem, the offsetting shape at each point is fixed and given. Therefore, we can calculate the offsetting shape more quickly with only one iteration.

### 3. CURED RESIN SHAPE FUNCTION

The shape function  $g(x, y, z)$  used in the non-uniform offsetting depends on the irradiance of a laser beam and the curing of liquid resin by the laser beam.

(1) The irradiation profile of an UltraViolet (UV) laser: Commercial SLA systems use a stationary UV laser with galvanometer-driven mirrors to scan a cross-section of a layer. The negative effect of the galvo scanner and the (optional) f-T lens on the geometric distortion of the scanned image is usually corrected by means of a software-based compensation. Sager and Rosen [2004] presented a dynamic laser beam model, in which, (a) laser beam size and shape may change according to location on surface; (b) irradiance profile may change depending on the point of focus; (c) laser beam angle with vertical axis changes during scanning; (d) refraction changes size, shape and location of cure profile.

(2) The curing model of liquid resin: Based on [Jacob 1992], the photopolymer resin obeys the Beer-Lambert law of exponential absorption. The laser irradiance distribution is Gaussian. That polymerization produces "bullet shaped" cured profiles. The models are only an approximation to the real situation and incorporate some assumptions, such as: (a) the flow of material due to convection or diffusion in any direction is negligible; (b) there are no light scattering, diffraction, refraction or reflection effects; (c) the heat generated is due to heat of polymerization only and the heat loss from the surface of the vat is negligible; (d) the evolution of the polymerization zone is unaffected by volume shrinkage; and (e) the evolution of polymerization is unaffected by oxygen inhibition. There are other researches who try to model the laser beam and how its intensity decreases with depth in a vat of resin. For instance, Brulle et al. [1994] used the Beer-Lambert Law to suggest an ideal model of laser-induced polymerization. Narahara and Saito [1992, 1993] suggested a mathematical representation based purely on the optical analysis of the laser beam path as it travels through the photosensitive resin, calculating the exposure distribution in the resin. Using this model, a solidified profile is determined by a threshold level of exposure.

In this paper, we assume the exact cure profile  $g(x, y, z)$  at various locations within a SLA platform is known to us. The function  $g(x, y, z)$  can be given either as an analytical model or a set of experimental data. Our non-uniform offsetting method is quite general. The proposed framework can take the shape function  $g(x, y, z)$  as either an analytical model or discrete sampling data. In this study, we will use a clipped ellipse as the shape function  $g(x, y, z)$  to test our method. An ellipse function is a good approximation to the "bullet shaped" cured profiles presented in [Jacobs 1992].

For the galvanometer-driven laser system, we will use a spherical coordinate system  $(r, \alpha, \phi)$  to represent any point  $\mathbf{v}(x, y, z)$  within the platform. Suppose the center of the platform at the curing plane is the origin  $\mathbf{O}$ . Therefore, the position of the mirror is at  $\mathbf{O}'(0, 0, L)$  (refer to Figure 1). For any point  $\mathbf{v}(x, y, z)$ , the main axis of the ellipse is  $\mathbf{v}-\mathbf{O}'$ . Suppose we define a new coordinate system at  $\mathbf{v}$  based on the axis  $\mathbf{v}-\mathbf{O}'$ . The three new axes are  $\mathbf{U}, \mathbf{V}, \mathbf{W}$ , which are defined as:

$$\mathbf{W} = \frac{(0,0,L) - (x,y,z)}{\|(0,0,L) - (x,y,z)\|}, \mathbf{V} = \frac{\mathbf{W} \times (1,0,0)}{\|\mathbf{W} \times (1,0,0)\|}, \mathbf{U} = \frac{\mathbf{V} \times \mathbf{W}}{\|\mathbf{V} \times \mathbf{W}\|},$$

Based on the axes  $\mathbf{U}, \mathbf{V}, \mathbf{W}$ , the ellipse at point  $\mathbf{v}$  can be defined as:

$$g(x, y, z) = v + a \cos(\theta) \sin(\phi) \mathbf{U} + b \cos(\theta) \cos(\phi) \mathbf{V} + c \sin(\theta) \mathbf{W}, z \leq 0.$$

where,  $a = b =$  radius of laser beam, and  $c =$  cure depth of liquid resin.

Suppose the radius of a laser beam is 0.005'' and the cure depth is 0.015''. Further suppose  $L = 13.7''$ . Some ellipse examples generated for various locations based on the above equations

are shown in Figure 4. Notice ellipse is symmetric around its center  $\mathbf{v}$ . Therefore it is quite easy to calculate  $(g(x, y, z))^{-1}$  from the sampling points of  $g(x, y, z)$ .

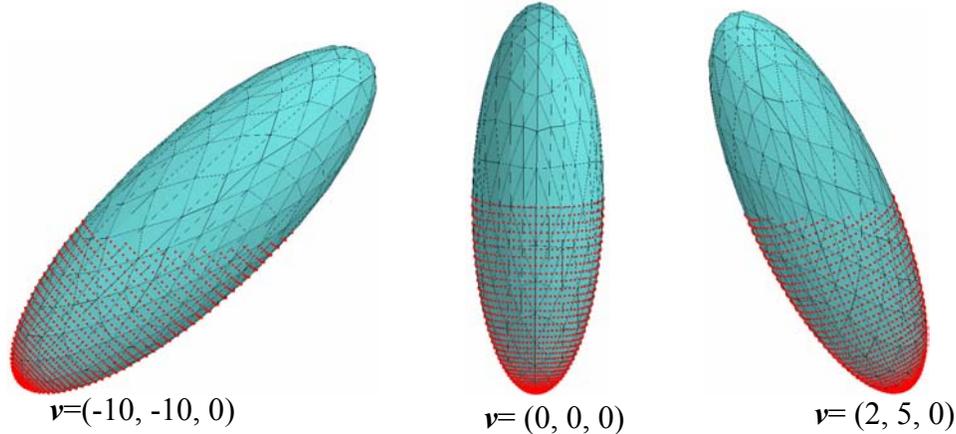


Figure 4: Cured resin shapes at three different locations.

#### 4. NON-UNIFORM OFFSETTING APPROACH

For a given geometry  $f(x, y, z)$  and a cured resin shape function  $g(x, y, z)$ , we need to develop an efficient and accurate algorithm to calculate the tool path  $h(x, y, z) = f(x, y, z) \oplus (g(x, y, z))^{-1}$ . Since a continuous object cannot be specified as a collection of finite number of points, such an object, in general, is specified by its boundary. This necessitates the computation of the boundary of the products  $\partial(A \oplus B)$  from the boundaries of the operands  $\partial A$  and  $\partial B$ .

The boundary,  $\partial(A \oplus B)$ , of the *Minkowski* sum of two sets  $A$  and  $B$  is a subset of the boundary of the *Minkowski* sum of their boundaries. That is,

$$\partial(A \oplus B) \subset \partial((\partial A) \oplus (\partial B)).$$

Therefore, we can calculate  $\partial((\partial A) \oplus (\partial B))$  first. Based on the generated results, we will then use a point-based method to clean up all the boundaries that are inside the offsetting distance and therefore do not belong to  $\partial(A \oplus B)$ . Finally we will use the remaining boundaries as  $\partial(A \oplus B)$  to reconstruct  $h(x, y, z)$ .

In the non-uniform offsetting problem, a difficulty is that  $\partial(g(x, y, z))^{-1}$  is constantly changing its shape within the platform. Consequently, it is very difficult, if possible, to calculate an exact solution of  $h(x, y, z)$ . We will calculate an approximation solution of  $h(x, y, z)$  instead.

Suppose  $g_1(x, y, z)$  and  $g_2(x, y, z)$  are the cured resin shape functions at two vertices  $\mathbf{v}_1$  and  $\mathbf{v}_2$ . We know: (1) if  $\mathbf{v}_1 = \mathbf{v}_2$ ,  $g_1 = g_2$ ; (2) if the distance  $\|\mathbf{v}_1 - \mathbf{v}_2\|$  increases, the difference  $(g_1 - g_2)$  also increases; (3) for any given tolerance  $\varepsilon$ , we can find  $\|\mathbf{v}_1 - \mathbf{v}_2\| < \alpha$  such that  $(g_1 - g_2) < \varepsilon$ . Therefore, within an error tolerance, we can use  $g(x, y, z)$  at vertex  $\mathbf{v}(x, y, z)$  to approximate all the shape functions at a small region around  $\mathbf{v}$ . This will significantly simplify the computation of  $\partial((\partial A) \oplus (\partial B))$  while it will still provide us with certain accuracy (less than an error bound  $\varepsilon$ ).

Therefore, our non-uniform offsetting method can be described as follows:

- (1) Based on a given error tolerance  $\varepsilon$  and laser shape function  $g(x, y, z)$ , we calculate a sampling resolution  $\alpha$  for  $f(x, y, z)$ ;
- (2) We use the calculated resolution  $\alpha$  to resample the given geometry  $f(x, y, z)$ ;
- (3) We calculate  $\partial(\partial f(x, y, z) \oplus \partial(g(x, y, z))^{-1})$  as the candidate offsetting boundary;

- (4) We use a point-based method to sample the candidate offsetting boundary and deleting all the sampling points that are inside the offsetting distance;
- (5) We reconstruct a valid mesh from the remaining points as the offsetting result  $\partial h(x, y, z)$ .

The remainder of this paper has been organized in the following manner. Section 5 presents a uniform resampling approach for any given mesh  $f(x, y, z)$  and a sampling resolution  $\alpha$ . Section 6 describes modeling operations to calculate the candidate offsetting boundary. Section 7 presents our approaches for sampling point generation and processing. The mesh reconstruction method is also briefly discussed in the section. Section 8 presents several examples of non-uniform offsetting results. Finally, Section 9 presents the conclusions and future research.

## 5. UNIFORM RESAMPLING OF INPUT MESHES

Mesh optimization methods, such as geometric and topological optimization based on vertex insertion and deletion, have been studied in lots of meshing literatures. They can be applied to offer improvements to mesh quality, even though the notion of mesh quality itself is an evolving concept [Shewchuk, 2002]. In our non-uniform offsetting method, a maximum mesh resolution  $\alpha$  is given such that any two points  $\mathbf{v}_1$  and  $\mathbf{v}_2$  within a triangle satisfy  $\|\mathbf{v}_1 - \mathbf{v}_2\| < \alpha$ . At the same time, we would like a minimum number of triangles to achieve the above condition, since the computational cost in the following steps increases with more triangles in  $f(x, y, z)$ . Therefore, for an input mesh  $f(x, y, z)$ , our goal is to find a resampling of the mesh such that we can use the minimum number of triangles to satisfy the mesh size control (that is, any two points  $\mathbf{v}_1$  and  $\mathbf{v}_2$  within a triangle satisfy  $\|\mathbf{v}_1 - \mathbf{v}_2\| < \alpha$ ).

We use a mesh optimization technique based on the concept of centroidal Voronoi tessellation [Du *et al.* 1999]. Given an open set  $\Omega \subseteq R^N$ , the set  $\{T_i\}_{i=1}^k$  is a tessellation of  $\Omega$  if  $T_i \cap T_j = \emptyset$  for  $i \neq j$  and  $\cup_{i=1}^k T_i = \bar{\Omega}$ . Given a set of point  $\{v_i\}_{i=1}^k$  belonging to  $\bar{\Omega}$ , the Voronoi region  $\hat{V}_i$  corresponding to the point  $v_i$  is defined by  $\hat{V}_i = \{x \in \Omega \mid \|x - v_i\| < \|x - v_j\| \text{ for } j = 1, \dots, k, j \neq i\}$ . The set  $\{v_i\}_{i=1}^k$  is a Voronoi diagram of  $\Omega$ , and each  $\hat{V}_i$  is referred to as the Voronoi region corresponding to  $v_i$ .

Given a region  $V \subseteq R^N$  and a density function  $\rho$  defined in  $V$ , the mass centroid  $v^*$  of  $V$  is defined by  $v^* = \frac{\int_V x \rho(y) dx}{\int_V \rho(y) dx}$ . Given points  $v_i, i = 1, \dots, k$ , the centroidal voronoi tessellation

$\{T_i\}_{i=1}^k$  is the one such that the Voronoi points are the same as the mass centroids of Voronoi regions (that is,  $v_i = v_i^*, i = 1, \dots, k$ ).

The centroidal Voronoi tessellation has a good property. Let energy is defined as

$$E(V_i, v_i) = \sum_{i=1}^k \int_{V_i} \rho(x) \|x - v_i\|^2 dx.$$

A necessary condition for  $E$  to be minimized is that the  $V_i$  are the Voronoi regions corresponding to the  $v_i$ , and the  $v_i$  are the centroid of the corresponding  $V_i$ . The convergence of the centroids was proved for the Euclidean metric under certain uniqueness assumptions [Du *et al.* 1999]. We use uniform density function  $\rho$  in this research, although we can use density functions such as geometric curvature to generate a mesh that is curvature based (that is, more triangles at high curvature areas and less triangles at flat areas).

We use *Lloyd's* method [Lloyd 1982] to calculate the centroidal Voronoi tessellation. It is a deterministic approach based on iterations of finding centroids and Voronoi regions, which is described as follows.

0. select an initial set of  $k$  points  $\{z_i\}_{i=1}^k$ ;
1. construct the Voronoi tessellation  $\{V_i\}_{i=1}^k$  of  $\Omega$  associated with the points  $\{z_i\}_{i=1}^k$ ;
2. compute the mass centroid of the Voronoi regions found in step 1; these centroids are the new set of points .
3. If this new set of points meets some convergence criterion such as maximum distance is less than a given tolerance, terminate; otherwise, return to step 1.

*Lloyd's* method is a fixed point iteration. We can calculate the number of points  $k$  based on the region area and the given mesh resolution  $\alpha$ . In addition, the mesh is constrained on the geometric boundary. One approach for the constrained centroidal Voronoi tessellation is to project the set of generators whose corresponding Voronoi regions contain a portion of the boundary of the domain onto the geometric boundary. In this research, we predetermine a subset of points along the boundary and fixed them at each iteration.

Therefore, for an input mesh  $f(x, y, z)$ , we take out each planar region and use *Lloyd's* method to generate its centroidal Voronoi tessellation. An example of a 2D region and the resampling result is shown in Figure 5. In the figure, randomly generated initial points and corresponding tessellation are shown in the left side; the calculated centroidal Voronoi tessellation for the region is shown in the right side. An example of a 3D model and the resampling result is shown in Figure 6. The input mesh is shown in Figure 6 (left) and the uniform resampling mesh is shown in Figure 6 (right).

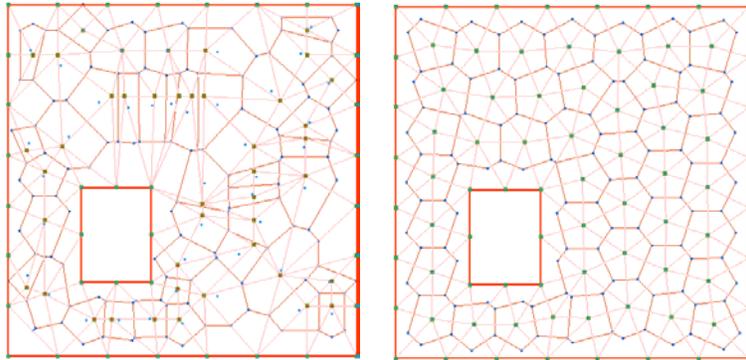


Figure 5: An uniform resampling example of an input 2D region.

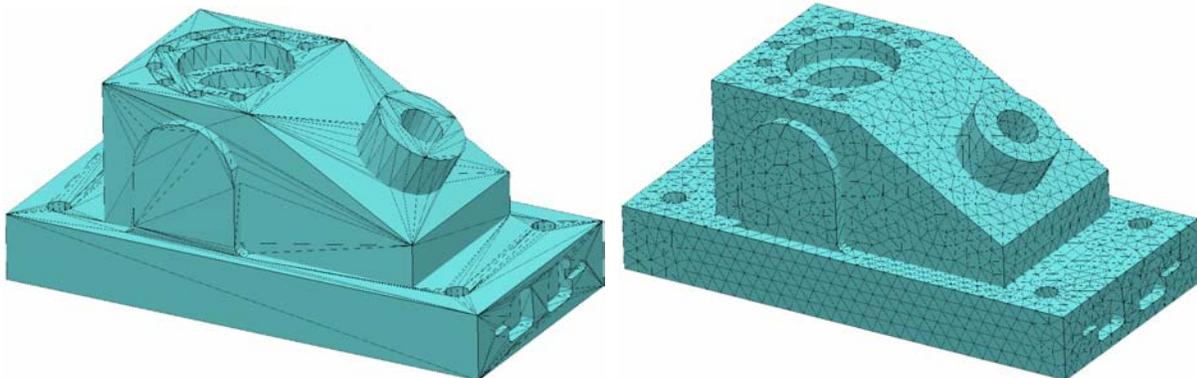


Figure 6: An uniform resampling example of an input 3D mesh.

## 6. TOOL PATH COMPENSATION OF VARIABLE SHAPES

For an input mesh  $S$ ,  $\partial S = V(S) \cup E(S) \cup F(S)$ , where  $V(S)$ ,  $E(S)$ , and  $F(S)$  refer respectively to the sets of all the vertices, edges and faces of  $S$ . Therefore, we will consider the offsetting of  $V(S)$ ,  $E(S)$ , and  $F(S)$  in order to calculate the candidate offsetting boundary.

### (1) Faces $F(S)$ :

For a triangle  $T \in F(S)$ , suppose a point  $p \in T$ . The corresponding laser shape function at  $p$  is  $g(x, y, z)$ . Suppose the offset point related to  $p$  is  $q$ . In a neighborhood of  $q$ , there is a parameterization  $q = q(u, v)$ , and distance  $\|p - q\|$  must reach a maximum along the triangle normal. Suppose the unit triangle normal at  $p$  is  $N$ . For the laser shape function defined as  $g(x, y, z) = v + a \cos(\theta) \sin(\phi) U + b \cos(\theta) \cos(\phi) V + c \sin(\theta) W$ , we can calculate the point  $q$  in order for  $\|p - q\|$  to reach a maximum value along  $N$  based on:

$$\frac{\partial(g(x, y, z) \cdot N)}{\partial \theta} = 0 \quad \text{and} \quad \frac{\partial(g(x, y, z) \cdot N)}{\partial \phi} = 0.$$

Therefore, we can calculate  $q$  by:

$$\tan(\phi) = \frac{U \cdot N}{V \cdot N}$$

$$\tan(\theta) = \frac{cW \cdot N}{a \sin(\phi) U \cdot N + b \cos(\phi) V \cdot N}$$

Notice we have a constraint  $z \leq 0$  for  $g(x, y, z)$ . Correspondingly, we have  $z \geq 0$  for  $(g(x, y, z))^{-1}$ . Therefore, if the calculated  $q$  based on the above equations has  $z < 0$ , we need to find a new  $\theta$  such that  $z_q = 0$ . This is the offset point for  $(g(x, y, z))^{-1}$  by considering the  $Z$  constraint. That is, the compensation component in  $Z$  direction is only required for down-facing triangles; for up-facing triangles, we only need to consider the compensation component in  $XY$  direction. This is illustrated in Figure 7.

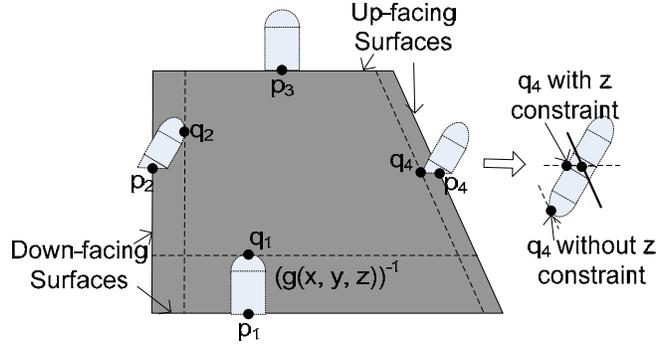


Figure 7: Different offsetting results for up-facing and down-facing triangles.

Suppose the three vertices of triangle  $T$  are  $p_1$ ,  $p_2$ , and  $p_3$ . Their corresponding offsetting points are  $q_1$ ,  $q_2$ , and  $q_3$  respectively. After resampling the input mesh, we know  $T$  is within a region with a radius less than  $\alpha$ . Therefore, we can approximate the offset result of  $T$  by a linear interpolating of  $q_1$ ,  $q_2$ , and  $q_3$ .

### (2) Edges $E(S)$ :

Suppose  $p \in E$  and the normals of two neighboring faces  $(f_1, f_2)$  of  $p$  are  $n_1, n_2$ . Suppose the offsetting point related to  $p$  is  $q$ . Parameterizing the curve in a neighborhood of  $q$  as  $q = q(t)$  and differentiating the  $\|p - q\|$  yields  $(p - q) \cdot \frac{\partial q}{\partial t} = 0$ . Therefore, for the corresponding laser shape function  $g(x, y, z)$ , we can calculate the offsetting point  $q_1$  and  $q_2$  corresponding to  $n_1$  and  $n_2$ . We then sample a set of normal points between  $n_1$  and  $n_2$  which are perpendicular to the edge. The calculated offset points form a curve on the surface of  $g(x, y, z)$ .

### (3) Vertices $V(S)$ :

Suppose  $q \in V$  and the neighboring edges of  $q$  are  $e_1, e_2, \dots, e_i$ . Suppose the offsetting point related to  $p$  is  $q$ .  $q$  lies in a region on the surface of  $g(x, y, z)$  which are formed by the offset curve of edges  $e_1 \sim e_i$ .

For an input mesh as shown in Figure 6, the generated candidate offsetting boundary for a simple sphere is shown in Figure 8. The offsets of  $F(S)$ ,  $E(S)$  and  $V(S)$  are illustrated in a close-up view. Notice not all the surfaces in the generated offsetting boundaries belong to  $\partial h(x, y, z)$ . Also the offsetting mesh does not form a valid STL model since there are lots of self-intersections. In the next section, we will present a point-based method to clean up the candidate offsetting boundaries and to reconstruct a valid offset mesh.

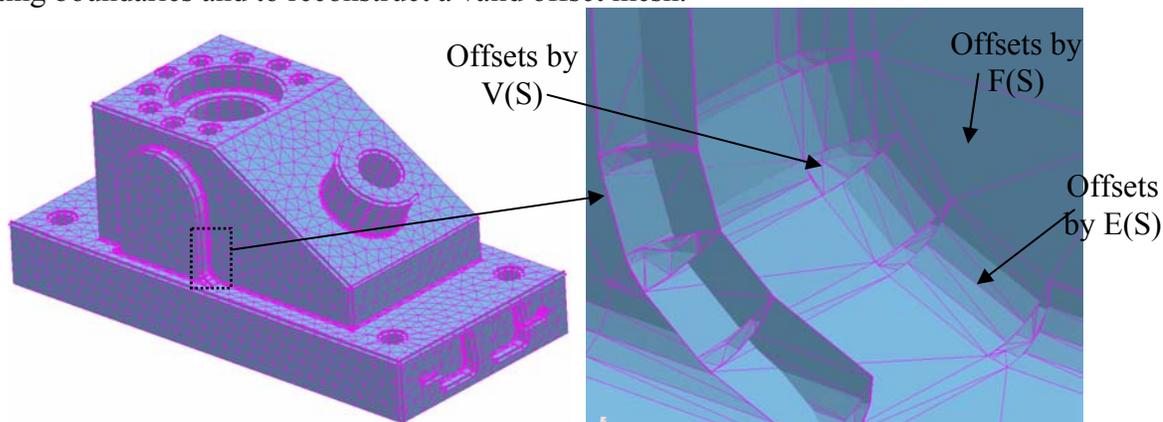


Figure 8: Candidate offsetting boundary for an input mesh.

## 7. TOOL PATH PROCESSING BASED ON SAMPLING POINTS

The candidate offsetting boundary,  $\partial((\partial A) \oplus (\partial B))$ , is a super set of the boundary of the *Minkowski* sum  $\partial(A \oplus B)$ . These candidate faces may intersect each other; also some of them are invalid for  $\partial(A \oplus B)$ . As shown in Figure 9, the faces that do not lie ON the boundary of the *Minkowski* sum will lie inside the sum. Therefore, we can judge if a small region of a face is valid based on if it is inside the *Minkowski* sum. If it is, we can set it as invalid for  $\partial(A \oplus B)$ ; otherwise, it lies ON the boundary of the *Minkowski* sum. However, it is difficult to directly trim the offset boundary due to the computational complexity and numeric instability during the trimming operation; it is also difficult to directly judge if a triangle lies ON the boundary of the *Minkowski* sum since the candidate offsetting boundary may have self-intersections.

In this research, we use a point-based approach for the classification of grid points and sampling points related to the boundary of the *Minkowski* sum [Chen *et al.* 2006]. We first generate a set of sampling points for geometric objects  $G_i$  of the candidate offsetting boundary. For each sampling point  $v_i$ , we calculate the related point  $v_k$  in the original model which we use to generate  $v_i$ . We construct a “stick” (a line segment  $v_i v_k$ ) and use it to check all the intersecting cells. We can label all the offset points as either too close to the original solid or possibly containing the offset surface. A 2D illustrative example is given in Figure 9. Offset surfaces of vertices  $v_1$  and  $v_2$  intersect

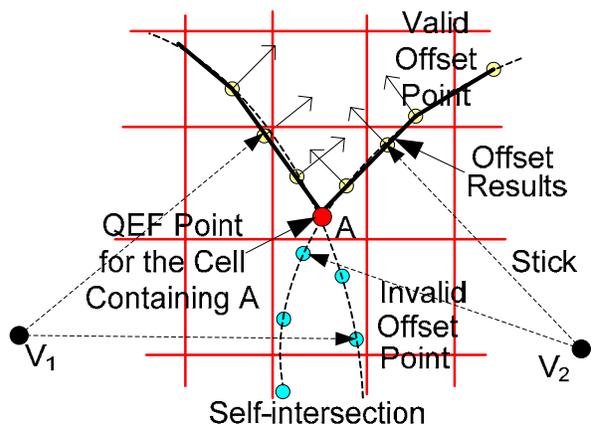


Figure 9: An illustration of our point-based offsetting method.

at point  $A$ . Surface regions below  $A$  are self-intersection. We use “sticks” to determine all the cells along the lines. In the figure, valid offset points are marked as yellow dots while invalid points (too close to  $v_1$  and  $v_2$ ) are marked as green dots. Detail algorithms are presented in [Chen *et al.* 2006].

After all the invalid offsetting points are deleted, we use a modified Dual Contouring method for reconstructing polygons from the remaining offsetting points. The major difference between our approach and the one given in [Ju *et al.* 2002] is the way of calculating a point in each cell for constructing polygons. That is, we calculate a QEF point based on the sampling points of geometric objects that intersect a cell; while in [Ju *et al.* 2002], a QEF point is calculated based on the intersection points (position and normal) of 12 cell edges with geometric objects.

Besides a uniform cell, we also construct an adaptively subdivided octree cell. By using the sub-divided cell, we can use a higher sampling rate to generate boundary points in a volumetric cell with small features or high curvatures. The approach for adaptive sampling of boundary cells is presented in [Chen 2007]. Based on the sampling points and volumetric cells, we can construct the isosurface of the volumetric grids as the offsetting boundary surface. Unlike the marching cube algorithm, Dual Contouring algorithm will not generate cracks for an adaptive grid with different grid sizes.

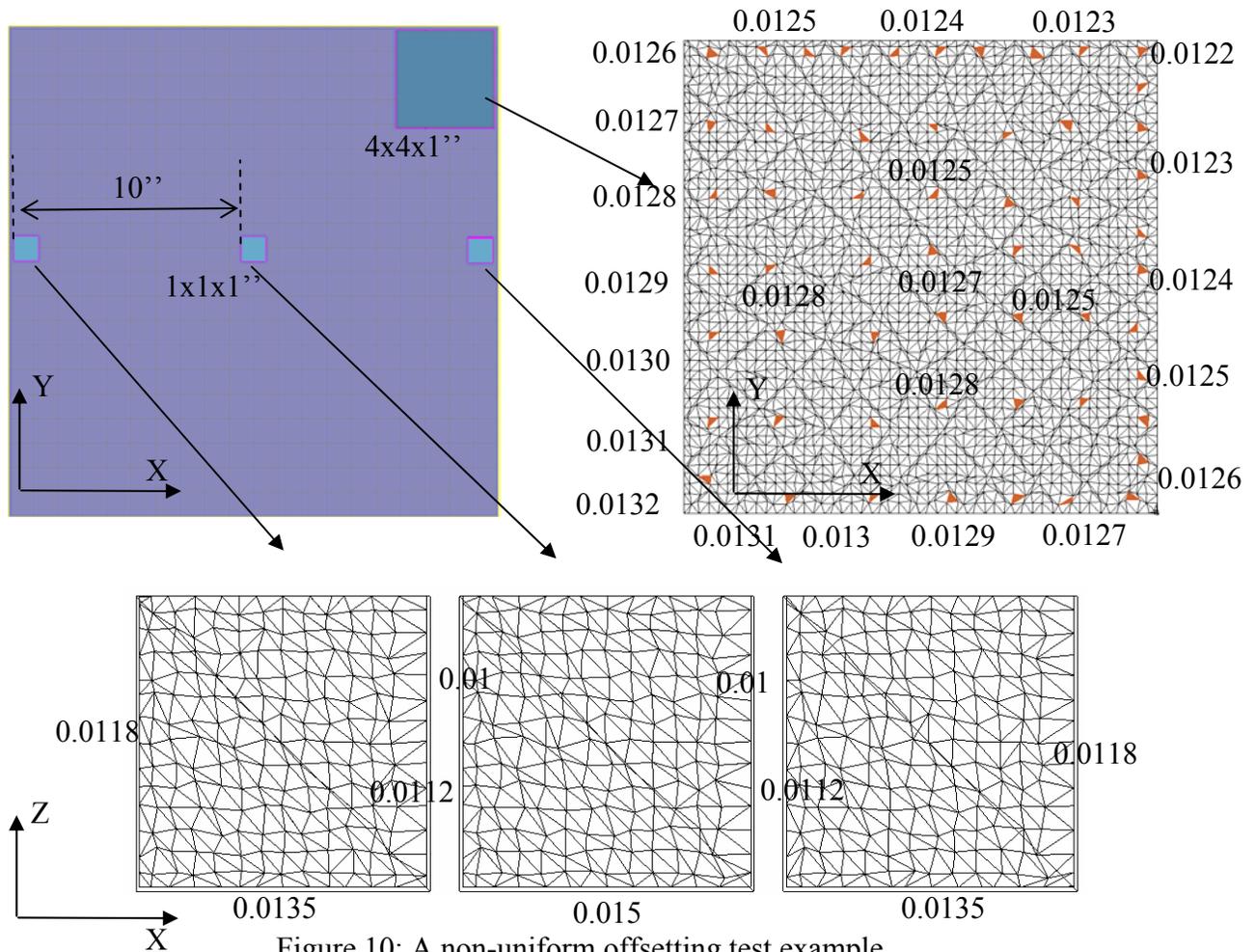


Figure 10: A non-uniform offsetting test example.

## 8. RESULTS AND DISCUSSIONS

We used C++ programming language with *Microsoft Visual C++* compiler to implement the presented algorithms. We have also tested our algorithms using a commodity PC with a 3.2 GHz *Pentium IV* processor and 2GB *DRAM* running *Windows XP*.

A test case is shown in Figure 10. In the test, we place four simple cubes in different locations within a platform as shown in Figure 10 (top left). For a 4''x4''x1'' cube at the top right corner, the bottom faces of the generated non-uniform offsetting result are shown in Figure 10 (top right). We select some triangles at various locations (shown as red triangles) and check their  $Z$  height. The  $Z$  values of the triangles vary from 0.0122'' ~ 0.0132''. The distribution of the  $Z$  values is shown in the figure. It is quite obvious the offset bottom surface is not flat any more due to the different cured resin shapes at different locations. However, by using them as the laser's tool path, the fabricated part will have a more flat bottom surface.

The offset results for three 1''x1''x1'' cubes are shown in Figure 10 (bottom). The offset distances for the bottom surfaces, the left and right vertical walls are shown in the figure. It is quite obvious that: (1) the offset distances for different faces of the same part are different; (2) the offset distances for the same face of the same part at different locations are different.

## 9. CONCLUSION AND FUTURE WORK

It is essential for a solid freeform fabrication (SFF) machine to have the capability of building parts with the same accuracy regardless of the building positions within its platform. This is required by all Rapid Manufacturing applications. In order to achieve the goal, we need to consider all kinds of systematic errors in the accuracy compensation including the one caused by different laser beam shapes. In the paper, we have presented a new non-uniform offsetting method for the tool path planning of SLA systems. Based on our approach, the generated tool path will consider various cured resin shapes within the platform. The test results on some simple geometry demonstrate the effectiveness of our method. The proposed non-uniform offsetting method will generate a compensated STL model from an input STL file. Since both  $Z$ -compensation and  $XY$ -compensation have been considered in our calculation, we can skip these steps in the slicing software of a SLA system and directly output slicing contours as the laser path. Therefore, our approach is very easy to be adopted in the current usage of SLA systems.

The non-uniform offsetting problem has much wider applications. For example, gantry-type CMMs suffer from "linear displacement" in which error margins grow as a sensor moves toward its center of its workspace along an axis. However, CMM control software can automate the offset compensation for these errors. As another example, the thermal distribution of a SLS platform is quite non-uniform. We are exploring the application of non-uniform offsetting to compensate the non-uniformity of the thermal distribution, and hence, to improve the accuracy of SLS system.

## REFERENCES

1. Brulle, Y., A. Bouchy, B. Valance and J.C. Andre. *Industrial photochemistry: chemical, transport and refractive index effects in space resolved laser photopolymerization*. Journal of photochemistry and photobiology A: Chemistry, 83 29-37, 1994.
2. Chen, Y., Wang, H., Rosen, D. and Rossignac, J. *A point-based offsetting method of polygonal meshes*. ASME Journal of computing and information science in engineering, in review, 2006.
3. Chen, Y. *An accurate sampling-based method for approximating geometry*. Computer-aided Design, accepted, 2007.
4. Du, Q., V. Faber, and M. Gunzburger. *Centroidal Voronoi Tessellations: Applications and Algorithms*. SIAM Review. Vol. 41, No. 4, pp. 637~676, 1999.

5. Farouki, R. T., H. P. Moon, and B. Ravani. *Minkowski Geometric Algebra of Complex Sets*. Geometriae Dedicata, Vol. 85, No. 1-3, pp 283-315, 2001.
6. Ghosh, P. *A Unified Computational Framework for Minkowski Operations*. Computer & Graphics, Vol. 17, No. 4, pp 357-378, 1993.
7. Jacobs, P.F. *Rapid Prototyping & Manufacturing: Fundamentals of Stereolithography*. 1992: Society of Manufacturing Engineers.
8. Jacobs, P.F. *Stereolithography and other RP&M Technologies: from Rapid Prototyping to Rapid Tooling*. 1996: Society of Manufacturing Engineers.
9. Ju, T., Losasso, F., Schaefer, S. and Warren, J. *Dual contouring of hermite data*. In *Proceedings of ACM SIGGRAPH 2002*, Computer Graphics Proceedings, Annual Conference Series, 339-346, 2002.
10. Kim, Y., G. Varadhan, M. Lin, and D. Manocha. *Fast Swept Volume Approximation of Complex Polyhedral Models*. Solid Modeling Conference 03, Seattle, WA, June 2003.
11. Lloyd, S. *Least square quantization in PCM*, IEEE Trans. Inform. Theory, 28 (1982), pp. 129-137.
12. Narahara, H. and K. Saito. *Fundamental analysis of a single layer created by three-dimensional photofabrication*. Proceedings of the Third International Conference on Rapid Prototyping, pp. 271-282, June 1992.
13. Narahara, H. and K. Saito. *Characterization of solidified resin created by three-dimensional photofabrication*. Proceedings of the Fourth International Conference on Rapid Prototyping, pp. 271-282, June 1993.
14. Osher, S., and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. 2003: Springer.
15. Rossignac, J.R. and A. A. Requicha. *Offsetting Operations in Solid Modelling*. Computer Aided Geometric Design, 1986. 3: p. 129-148.
16. Sager, B., and D. W. Rosen. *On the Use of Angled, Dynamic Laser Beams to Improve Stereolithography Surface Finish*. Proceedings of Solid Freeform Symposium 2004, Austin, TX, pp. 500 ~ 511.
17. Shewchuk, J. R. *What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures*. Eleventh International Meshing Roundtable (Ithaca, New York), pp. 115-126, Sandia National Laboratories, September 2002.