Build Style Decision Support for Stereolithography

Joel E. McClurkin and David W. Rosen
Rapid Prototyping and Manufacturing Institute
Georgia Institute of Technology
Atlanta, GA 30332-0405
(404) 894-9668 http://rpmi.marc.gatech.edu

ABSTRACT

When building parts in a stereolithography apparatus (SLA), the user is faced with many decisions regarding how the part will be built. The quality of the build can be controlled by the user via changing one of several build style variables, including part orientation, cross sectional layer thickness, and laser hatch density. A user will probably have preferences for the part build (i.e., accuracy or speed), but may not understand how to vary the build style variables to produce the desired results. A method based on response surface methodology and multi-objective decision support is described in this paper for relating build goals to three build style variables, and the use of these relationships in providing decision support for building a part on a SLA. The method is applied to the build style of a circuit breaker handle.

1. INTRODUCTION

Stereolithography (SLA) machines have dozens of processing variables that can be controlled by experienced operators to meet exacting build requirements of accuracy, surface finish, and build time. Experienced operators know qualitatively how process variables are related to build goals and can quantify some of their knowledge. Although SLA machine operation is predictable, the quality of resulting parts is not always obvious, particularly when trade-offs must be made among accuracy, surface finish, and build time.

A decision support method is presented in this paper to aid SLA operators in selecting appropriate values of process variables in order to achieve build goals as closely as possible. The goals can have importances specified in order to reflect preferences of the operator or part designer. The method is adapted from multiobjective optimization and utilizes response surface modeling to quantitatively relate variables to goals. Since our study was limited, we looked at only three process variables: part orientation, layer thickness, and hatch spacing, and three goals: accuracy, surface finish, and build time. Designer/operator preferences among the goals are also modeled using two different methods: weightings and priorities. Our objective is to render decision support by handling trade-offs among conflicting objectives quantitatively. Our method is demonstrated on a part with non-trivial geometries, a circuit breaker handle. Results reflect our intuition.

2. PREVIOUS RESEARCH

2.1 SLA Build Style Optimization

There have been several approaches to optimizing the build performance of SLA machines. These approaches vary from finding an optimal build layout to reduce build time to adaptively slicing a part to improve surface finish and reduce build time. One approach for optimizing build layout has been to use an expert system for determining preferred build direction as shown by Frank and Fadel [1]. Expert systems are heuristic in nature because most of the rule systems and knowledge used to create them are based on the experience of an expert. For example, an expert knows that thin layered parts build slower on the SLA, but are more accurate. This expert system uses information on geometric feature orientations to determine the preferred build style.

Another method for automating build layout addresses the problem of laying out a build with multiple parts to minimize build time. A problem arises when the parts need to be packed as closely as possible to maximize the utilization of space in the build vat and also reduce build time.

A genetic algorithm was developed by Wodziak et al. [2] to automatically place multiple parts to reduce build time. The parts may be translated or rotated +/- 90 degrees about the z-axis to aid in part packing. The creators of the genetic algorithms for part packing reported it to be a successful method for optimizing multiple part layout to reduce build time.

Build style optimization research has also been done with regards to build time and surface finish of curved surfaces. A technique, called adaptive slicing, involves slicing the part thinner in areas of high curvature and thicker throughout flat areas [3]. The main idea is to estimate the surface finish for a particular curved surface by determining how much stair stepping is present. First, the part is sliced using the maximum allowable layer thickness. If the amount of stair stepping creates a surface that is rougher than a pre-determined limit, the part is sliced thinner in that region. This work was done in the context of a fused deposition modeler (FDM).

A build style issue that is not dealt with in this paper, but is very important, is support structure generation. A method has been developed by Hur and Lee [4] to automatically generate support structures in SLA build styles. Not only does their method generate support structures, but it determines the best build orientation to minimize the amount of support structures required. Hur and Lee determine which facets of the STL file need to be supported based on the following criteria: orientation of the facet, area of the facet, part instability, and base support.

The build optimization research discussed in the previous paragraphs is based primarily on single objective optimization. For instance, the genetic algorithms for part packing utilized the single objective of reducing build time. The research in this paper deals with multiobjective decision support, where several objectives such as build time and accuracy are considered. A companion study was performed, based on selection among alternatives rather than multiobjective optimization, and was reported in [5].

2.2 The Compromise DSP

The method of rendering build style decision support described in this paper is based on solving a compromise Decision Support Problem (DSP) [6]. The compromise DSP is a type of multiobjective optimization method and is used to model decisions that consist of decision variables, constraints, and multiple goals. The goals are often in conflict with one another (i.e., accuracy and build time). The compromise DSP is an extension of Goal Programming and Sequential Linear Programming [7]. Solution of compromise DSPs is typically performed using the Adaptive Linear Programming (ALP) algorithm. The structure of the compromise DSP is shown below.

Given:	A feasible alternative, assumptions, parameter values.
Find:	Values of design and deviation variables.
Satisfy:	System Constraints, System Goals, and Bounds on variables.
Minimize:	Deviation Function that measures distance between goal targets and design point.

To use the compromise DSP, a part is presented in a "default" build style to serve the purpose of the existing alternative to be improved. This build style is then improved by changing the build style variables. The goals are build speed, part surface finish, and part accuracy. More accurate builds usually take longer to build. The alternative (in this case a build style) is improved by finding a combination of system variables such that all the system constraints are satisfied while the deviation function is minimized. The deviation function is solely a function of the deviation variables, which in turn are a measure of how well the system goals are met.

3. BUILD STYLE DECISION SUPPORT METHOD

Now that some of the background pertaining to build style decision support has been presented, the methods for solving the build style decision support problem are presented. This includes a brief discussion of the important build style variables and goals and how they are applied in the formulation of a compromise Decision Support Problem.

3.1 Build Style Variables, Parameters, and Goals

The build style variables, parameters, and goals are those quantities which drive the solution of the compromise DSP. Build style variables are those quantities which describe the attributes of the build style. These variables are directly affected by user input. Throughout the remainder of this paper, the build style variables will also be referred to as system variables. The system variables chosen for this project are the most common variables the user can change when setting up a build style. They are: Build Orientation (Surface Angle and Z-Height), Layer Thickness, and Hatch Spacing.

Build Orientation The build orientation variable is important because it can have an effect on every measure of build performance. The important aspects of the build orientation are the z-height of the part, and the various angles the surfaces of the part create with respect to the x-y plane. The part z-height can be controlled by the user by changing the orientation of the part. The bounds for this variable are 0.0025" and 10", which are limited by the smallest layer thickness and the size of the vat, respectively. However, the bounds of this variable can be narrowed significantly by dimensional information contained in the STL file that represents the part. Z-height can affect build speed dramatically. Builds with larger z-heights take longer to build because there are more build layers to build, causing more elevator movement and delay times.

Z-height also affects accuracy because the SLA is generally not as accurate in the z-direction due to overcure, print through, and build quantization errors [8]. Orientation also affects the locations of support structures which affect accuracy and surface finish. However, support structure generation and location is another issue too big to be dealt with in this paper. Technically build orientation can be considered a continuous variable due to the infinite number of orientations in which a part can be placed.

Layer Thickness When the STL file is sliced by the Maestro™ [9] processing software, it is sliced into many cross-sectional layers of finite thickness. The layer thickness variable refers to the thickness of these layers. The user has the ability to specify the thickness of the layers that represent the sliced part. The bounds for this variable are 0.0025" to 0.02". These bounds were determined from the Maestro™ build software version 1.8.

Layer thickness has a great effect on surface finish and accuracy, and to some extent build time. The accuracy and surface finish of curved surfaces will be better when thinner layers are used. However, parts usually build faster using thicker layers because recoat time will be greatly reduced. The phenomenon of build quantization plays a major role in accuracy and surface finish. If the top of a part falls on a half layer thickness, the SLA builds the extra layer which adds an extra half layer thickness to the top of the part. The scan speed of the laser, coupled with the pre-dip delay, affects down facing z-errors [8]. It is for the reasons above it is often necessary to select the *correct* value for layer thickness to increase build performance.

Hatch Spacing Hatch spacing is important because it affects build time, part accuracy, and surface finish. Hatch spacing refers to the density of the hatching pattern the SLA uses when "filling in" each cross section of the build. When the hatch vectors are closer together, the laser must scan more vectors per unit of cross sectional area, and the part will take longer to build. The fewer vectors the laser has to trace, the quicker the build will be completed. However, increased hatch vector density results in more resin being cured during the build and less during the post-cure process. This reduces part shrinkage, therefore increasing part accuracy. Also, as the hatch spacing on the up-facing surfaces (called fill vectors) increases, the quality of the surface finish decreases. If the hatch spacing is too wide, the build will most likely crash, which was experimentally observed. The bounds for hatch spacing were 0.004" to 0.020".

Build Style Parameters The build style parameters are quantities which describe properties inherent to the part being built. The build style parameters will also be referred to as system parameters throughout the rest of the paper. The user cannot change these properties because they are determined from the part's STL file or other uncontrollable parameters such as resin properties or laser power.

3.2 Modeling Of Build Goals

The build goals considered for this paper are: (1) Low Build Time, (2) High Accuracy, (3) Smooth Surface Finish. In order for the build style to be improved, it is necessary to be able to predict how the measures of build performance (the system goals) will change as the build style variables are altered. Response Surface Modeling is used to model the behavior of two goals, build time and accuracy, as a function of the build variables and parameters. An analytical equation is used to predict the behavior of the other goal, surface finish.

A response surface model is a means for expressing the build goal (the response) as a function of several build style variables. This is done by running several experiments and fitting a polynomial equation, called the response surface, to the resulting data. In support of this research, a series of experiments were conducted to generate data for the build time and accuracy response surfaces [10]. Four experimental factors were selected. A face-centered, central composite experiment design was used to create quadratic response surfaces. Block-shaped and cylindrical parts were built in various sizes and orientations to correspond to the experimental factors and their settings, shown in Table 1. Minitab™ is the statistical analysis software package that was used to define the experiments and fit the response surfaces.

Table 1 - Experiment Settings for each of the Build Style Variables

Build Style Variable	Low Value (-1)	Middle Value (0)	High Value (+1)
Layer Thickness (in.)	0.004"	0.006"	0.008"
Hatch Spacing (in.)	0.004"	0.012"	0.020"
Z-Height (in.)	0.5"	1.0"	1.5"
Part Volume	0.75 in^3	1.5 in^3	2.25 in^3

Accuracy Prediction The factors used to formulate the accuracy response surface are: part height, layer thickness, and hatch spacing. Part height was chosen as a factor because the SLA is less accurate in the z-dimension than in the x and y-dimensions. Layer thickness affects build quantization errors, which detract from accuracy. Hatch spacing has a dramatic affect on post cure shrinkage which in turn greatly affects accuracy. When the hatch spacing is less dense more resin is cured after the build process and this causes shrinkage.

The unrefined response surface for accuracy is shown in Eqn. 1. The fit parameters are: S = 380.0 and R-Sq = 92.1%.

$$\begin{aligned} Accuracy &= -12.6E6(H)(LT) + 11125(Z)(LT) + 48344(Z)(H) + 96.2E6(LT)^2 \\ &+ 3470000(H)^2 + 577(Z)^2 - 703617(LT) - 41629(H) - 321(Z) + 2665 \end{aligned} \tag{1}$$

Build Time Prediction The Build Time Estimator (BTE) program [11] was used to obtain data for the build time response surface, rather than building each parts individually. All four experimental factors from Table 1 were used. The BTE program is usually accurate within 3 percent.

The unrefined build time response equation is shown in Eqn. 2. The statistical fit parameters are: S = 48.61, R-Sq = 95.4%.

$$\begin{aligned} Build\ Time &= 117(V)(Z) - 203(H)(Z) - 5135(H)(V) - 33813(LT)(Z) + 1125(LT)(V) \\ &- 1910000(LT)(H) + 95(Z)^2 + 42(V)^2 + 693238(H)^2 + 9470000(LT)^2 + \\ &33(Z) - 81(V) - 4479(H) - 70138(LT) + 301 \end{aligned} \tag{2}$$

where: V = part volume (in^3), Z = part height (in), H = hatch spacing (in), and LT = layer thickness (in).

The resulting equations (Eqns. 1 and 2) are used to define the build style aspiration space which determines the performance of a particular combination of build style variables. These equations can be refined to eliminate some statistically insignificant variables, however this could reduce the accuracy of the equation with minimal benefit to computational time.

Surface Finish Prediction The theoretical method for determining surface finish uses equations based on layer thickness and the surface angle. The surface angles can be determined from the STL file that describes the part. In order to predict the surface finish for a given STL file, a computer routine to estimate surface finish is used. This routine takes uses the STL file as the input. From this information, the routine can predict the *relative* surface finish of several different build orientations. The equation used to predict surface finish for one facet is shown in Eqn. 3 and was calibrated using experimental data.

$$PRC = (2 * \cos(\Theta) * \sin(\Theta) * 937) + 3.5 * \Theta + 48 \qquad (3)$$

The predicted surface finish will be weighted by facet area, then normalized for use in the compromise DSP formulation.

3.3 Compromise DSP Formulation

As mentioned in Section 2, the compromise DSP is a design decision making formulation. Values for a set of design variables are chosen to meet a set of goals as well as possible, subject to constraints. For our problem, the design variables are the build style variables, part orientation, layer thickness, and hatch spacing. Goals include build time, accuracy, and surface finish. The constraints for build style decision making apply to all builds; they arise due to the SLA machine limitations and manifest themselves as bounds on the design variables.

For a compromise DSP, goals are formulated using Eqn. 4. In this formulation, the objective is to bring each of the goals to the minimum achievable value for that goal. That is, it is desirable to have the numbers quantifying build time, accuracy, and surface finish as close to the minimum as possible. This requires estimating the minimum (best) and the maximum (worst) possible values for each of the system goals. This normalizes the goals to ensure that no single goal dominates the solution process. The closer the goal value is to 0, the smaller the deviation variable will be. For this formulation the deviation variable d_i^- will always be zero because it is impossible to underachieve a minimum. Thus, the only deviation variable under consideration is d_i^+.

$$\frac{A_i(X) - A_{i,\min}(X)}{A_{i\max}(X) - A_{i,\min}(X)} + d_i^- - d_i^+ = 0 \qquad (4)$$

where: $A_i(X)$ = The achievement function, $A_{i,\min}(X)$ = the minimum (best) possible achievement, $A_{i,\max}(X)$ = the maximum (worst) possible achievement. The minimum and maximum values are dependent on the part being built. The equations for $A_i(X)$ were given as Eqns. 1, 2, 3 in the previous subsection. X represents the vector of build variables.

A deviation function measures the distance from a point in the feasible design space to the goals. Deviation functions can be formulated in two ways: the preemptive formulation which uses goal priorities, and the Archimedian formulation which uses goal weights (importance values). Using the compromise DSP, the mathematical formulation of the build style decision problem for the circuit breaker handle in Section 4 is shown in Figure 1.

Solving compromise DSP problems is usually accomplished using a multiobjective optimization code such as DSIDES [6]. However, since the build style design space is relatively small, an exhaustive search (i.e., every possible combination of build style variables) can be performed to yield a solution fairly quickly. With this method, the true minimum value of the deviation is found in about two seconds. The ranges and number of values for each design variable are listed here along with the number of points in parentheses: Layer Thickness 0.002" - 0.010" (17), Hatch Spacing 0.004" - 0.020" (17), and Part Height 0" - 10" (3). Height is dependent on three different build orientations, defined by the dimensions of a bounding box, but could be increased to many more orientations to account for support structures or other build goals not considered in this paper. Thus, the number of possible points to evaluate for any part is 17 x 17 x 3 = 867.

Given: STL file of circuit breaker handle. Goal importances.

Find: Values of system variables: X_1 = layer thickness, X_2 = hatch spacing, X_3 = part orientation
Values of the deviation variables associated with the goals:
d_1^+ - build time, d_2^+ - accuracy, d_3^+ - surface finish

Satisfy: System goals:

$$\frac{A_1(X)-116.8}{1006.6-116.8}-d_1^+=0 \qquad \text{Build Time}$$

$$\frac{A_2(X)-2088}{8725-2088}-d_2^+=0 \qquad \text{Accuracy}$$

$$\frac{A_3(X)-251.3}{523.2-251.3}-d_3^+=0 \qquad \text{Surface Finish}$$

Bounds: Layer Thickness $0.002'' <= X_1 <= 0.010''$
Hatch Spacing $0.004'' <= X_2 <= 0.020''$
Build Height $0'' <= X_3 <= 10''$

Minimize: the Deviation Function:
$Z = [d_3^+, 0.5(d_1^+ + d_2^+)]$ (Scenario 1), $Z = \Sigma W_i d_i^+$ (Scenarios 2, 3)

Figure 1 - Compromise DSP Build Style Problem Formulation.

The exhaustive search method has been embodied in a software tool called CABSS (Computer-Aided Build Style Support) and has been validated using several examples, one of which is included in the next section.

4. A TEST CASE: CIRCUIT BREAKER HANDLE

This section shows some results of applying the compromise DSP to generate build styles for different solution scenarios. The part considered is a circuit breaker handle manufactured by the Siemens corporation. This part was chosen for its geometric complexity. The breaker handle was scaled to twice its normal size so that it would lie within the bounds of the experimental design space that was used to create the response surface models. When using empirically derived equations, it is important to only interpolate within the bounds of the experiment space, and not extrapolate beyond these boundaries. The wireframe and STL models of the handle are shown in Figure 2. Its overall dimensions are 1.45" x 1.62" x 2.86" (X x Y x Z).

4.1 Solution Scenarios

The problem of determining the build style will be solved using three different scenarios for prioritizing and weighting the build goals. Scenario 1 emphasizes surface finish using a preemptive formulation of the deviation function. Scenario 2 places a high importance on accuracy. Scenario 3 places a medium emphasis on build time, followed by accuracy. Numerical weighting values for these scenarios are shown below. These scenarios will be used to determine how the build style variables change with respect to changing build goals.

Goal	Scenario 1	Scenario 2	Scenario 3
Surface Finish	Level 1, Weight 1.0	0.1	0.2
Build Time	Level 2, Weight 0.5	0.2	0.5
Accuracy	Level 2, Weight 0.5	0.7	0.3

4.2 Results

The STL file of the handle was entered into CABSS where volume, surface area, and theoretical surface finish for three orientations were calculated. After entering the build scenarios (above), CABSS exhaustively searches the design space defined by goal equations presented in Section 3. The results of this search are presented in Table 2.

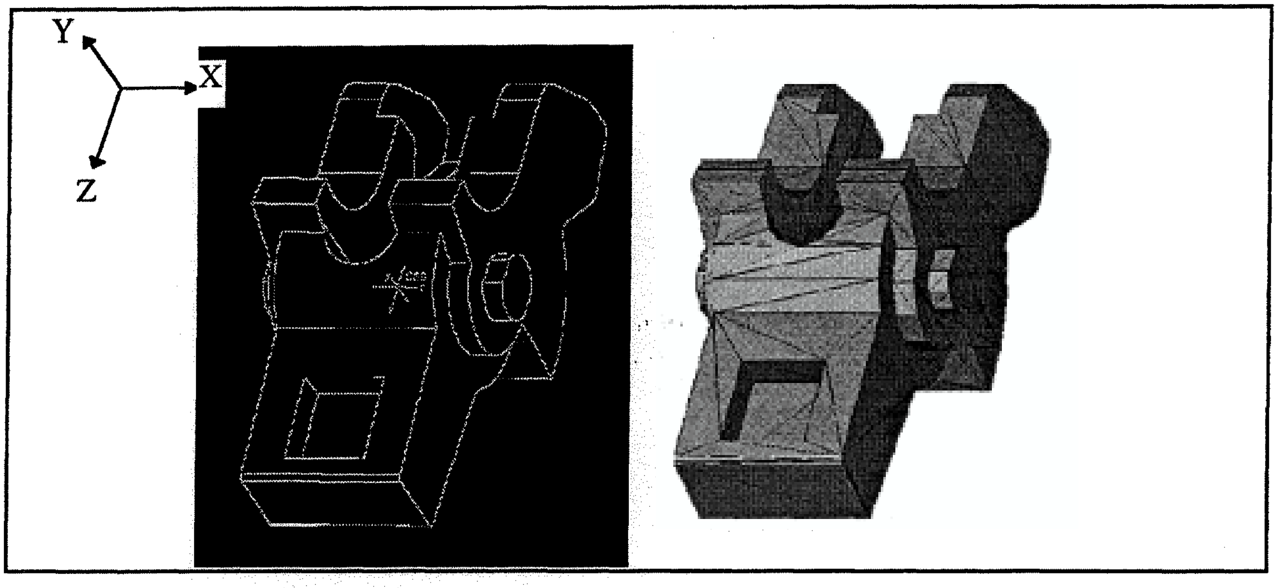

Figure 2 - Siemens circuit breaker handle. Wire frame and STL renderings

Table 2 - Results from Exhaustive Search for Circuit Breaker Example

Scenario	Build Style Variables	Build Goals	Deviation Function
#1	Layer Thickness = 0.0055" Hatch Spacing = 0.013" Height = 1.45"	Build Time = 301.28 Average Inaccuracy = 2597.21 Surface Finish = 251.77	level 1 = 0.0 level 2 = 0.061
#2	Layer Thickness = 0.0045" Hatch Spacing = 0.007" Height = 1.45"	Build Time = 403.7 Average Inaccuracy = 2239.48 Surface Finish = 251.77	level 1 = 0.036
#3	Layer Thickness = 0.006" Hatch Spacing =0.015" Height = 1.45"	Build Time = 271.74 Average Inaccuracy = 2824.78 Surface Finish = 251.77	level 1 = 0.046

From knowledge of SLA operation, build time decreases as layer thickness and hatch spacing increase. However, part inaccuracy increases as layer thickness and hatch spacing increase. Therefore, it is reasonable to assume that as the importance of accuracy increases, layer thickness and hatch spacing should decrease. Increasing the importance of build time should have the opposite effect on these values. As can be seen from Table 2, layer thickness and hatch spacing attain their largest values in Scenario 3, when build time is most important. In Scenario 2, when accuracy is most important, layer thickness and hatch spacing take on their smallest values.

In order to validate the results obtained, the handle should be built with the "best" build style variable values. At present, only the build time goal has been tested. The BTE program provided the baseline build time measurements. Comparison of BTE and CABSS build times shows an error of less than 2 percent for Scenarios 1 and 3 and an error of 5 percent for Scenario 2.

5. CONCLUSIONS

Several conclusions can be drawn from this research. The main conclusion is that it is indeed possible to provide decision support for stereolithography build styles using the compromise DSP. This is done by formulating and solving a DSP with a computer tool. Several specific conclusions are used to support this main conclusion.

- The selected build style variables have a significant impact on SLA build performance: build orientation (part height and surface angles), cross sectional layer thickness, and laser hatch density. This research verified the expected behavior of these variables.
- Build performance can be predicted by constructing response surface models of the build goals as functions of the relevant build style variables and parameters. These response surface models apply to parts of a wide range of geometric complexities and sizes. This conclusion is based on observing the test case results and comparing them with expected SLA build behaviors. However, the response surface models should only be applied to parts whose parameters fit within the experimental design space on which the response surface is based.

Even though this method for decision support is viable, more work needs to be done to improve the results. This includes creating more accurate response surface models that encompass a wider range of part parameters (volume and height).

ACKNOWLEDGEMENTS

We gratefully acknowledge the support from ARPA TRP grant #2187 and from the RPMI member companies.

REFERENCES

1 Frank, D and Fadel, G, "Expert System Based Selection of the Preferred Direction of Build for Rapid Prototyping Processes," Clemson University report, 1993.

2 Wodziak, J R, Fadel, G M, Kirschman, C, "A Genetic Algorithm for Optimizing Multiple Part Placement to Reduce Build Time," Clemson Intelligent Design Systems Laboratory, Clemson University, 1994.

3 Sabourin, E, Houser, S A, Bøhn, J H, "Adaptive Slicing using Stepwise Uniform Refinement," *Rapid Prototyping Journal*, 2(4):20-26, 1996.

4 Hur, J and Lee, K, "Efficient Algorithms for Automatic Support Structure Generation in Layered Manufacturing," Proceedings ASME Computers in Engineering Conference, paper #96-DETC/CIE-1324, 1996.

5 McClurkin, J E and Rosen, D W, "A Build Style Selection Method for Stereolithography," Proceedings 1997 NASUG Conference, Orlando, February 16-20, 1997.

6 Mistree, F, Hughes, O F, Bras, B A, "The Compromise Decision Support Problem and Adaptive Linear Programming Algorithm," *Structural Optimization: Status and Promise*, AIAA, Washington, DC, pp. 247-289, 1993.

7 Mistree, F, Hughes, O F, Phuoc, H B, "An Optimization Method for the Design of Large, Highly Constrained Complex Systems," *Engineering Optimization*, 5(3):141-144, 1981.

8 Tata, K and Flynn, D, "Quantification of Down Facing Z Error and Associated Problems," Proceedings 1996 NASUG Conference, San Diego, March 11-13, 1996.

9 Maestro 1.8 Users Manual, 3D Systems Inc., Valencia, CA, 1996.

10 McClurkin, J E, "A Computer-Aided Build Style Decision Support Method for Stereolithography," Masters Thesis, Georgia Institute of Technology, June 1997.

11 McClurkin, J E, Build Time Estimation via the World Wide Web, available at: http://rpmi.marc.gatech.edu/BTE.html.