Advanced Process Planning for Subtractive Rapid Prototyping

Joseph E. Petrzelka and Matthew C. Frank
Department of Industrial and Manufacturing Systems Engineering
The Iowa State University of Science and Technology
Ames, Iowa, USA

Abstract

This paper presents process planning methods for Subtractive Rapid Prototyping, which deals with multiple setup operations and the related issues of stock material management. Subtractive Rapid Prototyping (SRP) borrows from additive rapid prototyping technologies by using 2½D layer based toolpath processing; however, it is limited by tool accessibility. To counter the accessibility problem, SRP systems (such as desktop milling machines) employ a rotary fourth axis to provide more complete surface coverage. However, layer-based removal processing from different rotary positions can be inefficient due to double-coverage of certain volumes. This paper presents a method that employs STL models of the in-process stock material generated from slices of the part along the rotation axis. The developed algorithms intend to improve the efficiency and reliability of these multiple layer-based removal steps for rapid manufacturing.

Introduction

Subtractive Rapid Prototyping (SRP) is a considerably lesser known and utilized form of rapid prototyping technology, mainly due to continued challenges in the pre-process engineering and setup planning required. Subtractive operations in general afford excellent accuracy and repeatability, which is why it is more often utilized as an additional function of a hybrid-type rapid prototyping system that includes both additive and subtractive operations. Since the early days of RP technology, with systems such as Laminated Object Manufacturing (LOM), subtractive means have been employed as part of a solution alongside the conventional additive layer-based approaches. Although perhaps not purely subtractive, in LOM, a laser was used to cut the profile and surrounding support structures of each layer, after the layer of paper is added. In this manner, the additive approach enabled improved geometric capability while the laser cutting offered reasonable shaping accuracy. Later, the Sanders Modelmaker system, now used heavily in jewelry and dental manufacturing, utilized a machining process to accurately mill each deposited layer to a precise thickness. In a research project, Shape Deposition Manufacturing (SDM) used 5-axis machining in conjunction with a variety of deposition approaches to create both parts and molds. More recently, systems such as LAMP and Ultrasonic Consolidation (UC) continue to use subtractive means in an iterative manner to improve accuracy and surface finish. Desktop milling machines suitable for rapid prototyping have been marketed for lower-end materials in single or multi-sided machining operations, while lower-end, but more user-friendly SRP software systems such as Millit and Deskproto have attempted to facilitate the NC programming for these applications. To date, subtractive processes continue to be utilized in a few RP systems, while SRP-only systems have had limited success. The overwhelming success of RP technologies to date can be attributed almost solely to the additive-only machines; however, there are niche applications where an SRP system would be desirable. This paper presents one of several new methods the authors have developed for SRP process planning, specifically one that will enable more efficient execution and reduced processing times.

Subtractive processes in general are extremely versatile and can form a variety of complex geometries from a vast array of materials. However, to form arbitrary geometry from stock, material must be removed from a variety of orientations, or setups. For example, if one were to mill a part from a block of stock material, several setups would be required; one possibility, though not necessarily optimal, would be to mill the part from orthogonal angles (Figure 1). In situations such as this, it becomes very difficult to analyze the remaining stock after a certain sequence of operations. The problem is far from trivial; component geometry will often occlude accessibility to certain portions of the stock material. A variety of tool path simulation and verification techniques have been presented in the literature; however, these techniques focus computational effort on surface finish and conformity rather than gross shape analysis.

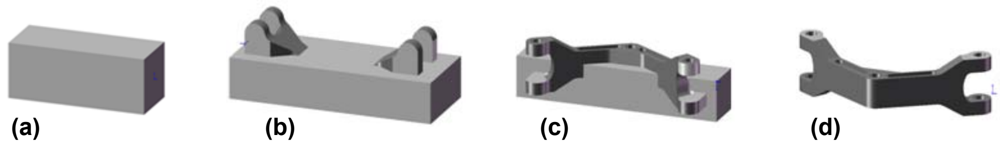


Figure 1 – Multiple setups required (even in multi-axis milling) to create component via a subtractive process; (a) shows initial stock geometry and (b-d) show the results after iterative cutting operations until (d) the component fully emerges

Little research has been conducted that specifically evaluates the effect of multiple-setup roughing operations; in fact, many typical situations do not require this type of analysis. In usual production settings, a near-net shape component is created (i.e. casting), and then critical features are machined. In die and mold shops, a single setup in a multi-axis mill is typically sufficient to achieve the geometry of the die or mold cavity. Regardless, there exist a variety of applications where multiple setup subtractive analysis is a requirement, especially for creating custom or unique parts from nominally sized stock material. In this research, we consider a subset of this problem where all setups can be performed by rotation about a single axis. The manner of iteration is not important; the work piece could be fixtured in a four-axis machine and automatically rotated to new orientations, or the work piece could be manually fixtured in a three-axis mill such that each subsequent orientation was about a theoretical axis (as in Figure 1).

Recent development of a rapid prototyping process using CNC machining by the authors [8] has enabled automated process planning for custom, very low volume, or prototype components. In this system, called CNC-RP, sacrificial support structures provide fixturing for machining in a four-axis milling machine. Visibility analysis of cross sectional slice data provides a basis for automated setup planning about a single axis. The implementation uses a modified Greedy set cover algorithm to determine discrete orientations; depth requirements are generated by 'bucketing' visible slice segments for each orientation. In its full implementation, CNC-RP safely and completely machines arbitrary components from bar stock in a four-axis mill. In effect, the fourth axis serves as a rotary indexer to automate setups between independent three-axis tool paths (Figure2).

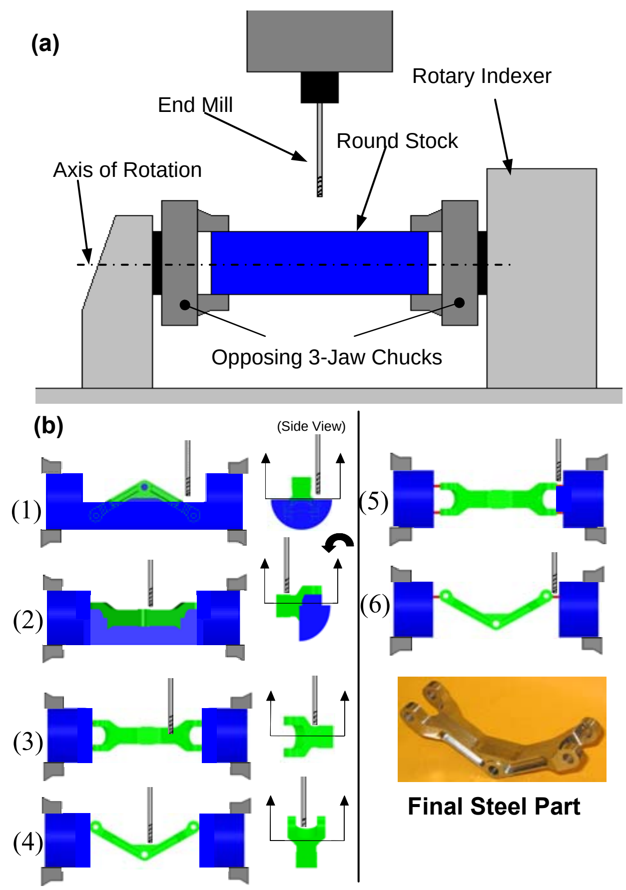


Figure 2 - (a) CNC-RP Implementation; (b) Process sequence of steps (b.1-b.4) to expose component geometry and (b.5-b.6) to expose sacrificial supports

While this system has proven capable and robust, it generates each tool path independently of others, basing it only on set cover angles and depth results. The conservative tool path planning method results in efficiency sacrifices for this multiple orientation process. Because each machining orientation is performed independent of others, there is no knowledge of prior operations during roughing procedures. This naïve approach leads to excessively long tool paths and redundant volume sweeps (Figure 3). If an effective means existed to predict the stock geometry resulting from each step (setup) in the process, one could generate much more effective and efficient tool paths. Saito [18] cites machining time as one of five critical factors to successfully producing complex geometries; naïve tool path generation fails to minimize this factor.

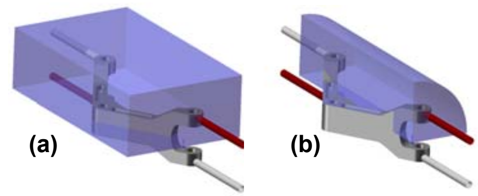


Figure 3 - (a) Naïve tool path volume sweep; (b) Volume sweep of tool path with knowledge of prior operations

To characterize the tradeoff between robust (naïve) and fast characteristics, we define two types of error that may occur when generating the in-process stock material shape; *Type A*: Remaining stock is underestimated, where the failure mode is possible collision between tool and large amount of stock, and, *Type B*: Remaining stock is overestimated, where the failure mode is a marginal increase in tool path length and machining time. Noting the failure modes and the desire to develop a robust system, it is asserted that *Type A* error must be avoided at all costs. For example, in the lights-out operation of a Subtractive Rapid Prototyping method like CNC-RP, it is absolutely necessary that tool crashes do not occur during untended operations, often done overnight. Without simulation of previous operations (three axis tool paths at different orientations), each subsequent tool path is required to assume worst-case conditions (Type B error). While little original stock material may actually remain, a large volume is swept by the tool to maintain a robust process (Figure 3a). Commercially available CAM packages can accept STL (triangle mesh surface) models as stock material input to constrain tool path generation to the correct regions in space. Attempts to utilize existing simulation techniques in CAM packages to generate this driving STL model have not been successful. Z-buffer based simulations provide reasonably accurate results, though not robust. After several iterations between the Z-buffer dexel structure to a portable STL format (for use simulating several sequential machining operations), the data becomes unreasonably large and prone to corrupt STL files. Furthermore, this and similar simulation techniques rely on pre-generated tool paths. These difficulties create the need for a simple remaining stock calculation for multiple setup environments that is (1) mathematically defined, with portable results, (2) computationally inexpensive for application to many unique components, and (3) able to predict process performance prior to computationally expensive tool path planning.

Related Work

Algorithms for subtractive process planning can be classified as (1) tool path generation methods or (2) simulation / verification methods [18]. The literature fails to address the area of 'lightweight' pre-NC computation (gross shape analysis only, neglecting surface scallops) for process planning in a multiple-setup environment. Tool path simulation is similar to the remaining stock calculation desired; it creates a model of the stock by subtracting the tool's swept volume from a specific tool path. Verification techniques go a step beyond to detect gouges (over-cutting during feed motion), under-cutting (failing to remove all material from a component surface), and collisions (driving to tool into a part or fixture during rapid travel). By definition, simulation and verification calculate (in various manners) the Boolean effect of a tool path. Three primary groups of verification / simulation are evident in the literature: vector based, spatial discretization, and solid/parametric surface based methods. Chappel [5] and Oliver/Goodman [16] developed vector based approaches to track machining progress. After populating component surfaces with normal vectors (the density determined relative accuracy and computation time, though in O(n)), the vectors are trimmed as appropriate for each tool instance. Jerard [12] extended this simulation technique to verification and cutter location (CL) data correction. The most popular spatial discretization method – the Z-buffer - was originally proposed by Van Hook [19]. Huang and Oliver [11] extended this simulation technique to verification by detecting gouging and undercutting. Saito and Takahashi [18] presented a similar graphics-based approach (G-buffer method) that is capable of both tool path generation and simulation. Bohez et al. [3] discretize space into x-y planes in their method. Solid modeling would seem to be ideal for NC simulation and verification; an exact representation of remaining material could be constructed. However, constructive solid geometry (CSG) application becomes $O(n^4)$ and therefore unacceptable for any simulation of practical scale [12]. There does not appear to be extensive recent research in this area; most efforts have been focused on parametric boundary representation (Brep) methods (effectively a transition from solid to surface modeling). Using Brep methods, Weinart [21] achieves 5-axis swept volume approximation resulting in parametric NURBS surfaces. Boolean subtraction of this swept volume from a stock model results in a tool path simulation. Weinart still interpolates between discrete tool instances (similar to discretization methods) and shows only a small example with no large scale implementation. Jütler [13] presents a similar method, though he used discrete points along a rational B-spline curve to drive tool instances. Jütler presents purely empirical research with no successful implementation. Blackmore [2] generates triangle mesh representations of swept volume using differential equations applied to continuous tool paths and boundaries. All of these Brep methods are slow [3] and do not appear to be mature technologies based on the lack of full scale implementations in the literature. Lauwers [15] and Huang [11] both use simulation results to correct CL data (tool position & posture). Pal [17] presents a remaining stock computation to drive more efficient tool paths, but his algorithm is designed only for two-stage three axis milling with sequentially smaller ball mills. Many of these simulation techniques are well suited to verify gouging and undercutting in surface finishing operations. However, most are very slow (Z-buffer methods are the only that can be considered fast [3]) and all rely on computational investment in tool path generation. The literature does not present methods suitable for gross stock remainder calculations for more efficient roughing processes. Only vector based approaches offer mathematically defined tool tracking with respect to the surface (though some other methods can derive it from discretized maps).

Methodology

In this section, we provide a general description of the proposed methods for remaining stock calculation. The approach involves three major tasks; (1) the model of the stock material is broken into cross sectional slices and a set of operations performed to create a factor of safety for avoiding *Type A* errors, (2) slices are modified to simulate the iterative changes to the stock, and finally (3) a reconstruction of the slice data into a surface model representation of the stock for each iterative step in the subtractive rapid prototyping process. The method presented computes the remaining stock for a set of subtractive processes where all setups occur about a single theoretical axis. In particular, this analysis computes the net effect of a tool path from a given orientation to a given depth, but without requiring exact tool path data. If a robust tool path generator is used, the only gain from considering tool instances is the ability to analyze surface scallops; for a roughing operation this is considerably less important than in finishing operations. Our method consists of three distinct algorithms: slice approximation, slice shadowing, and polyhedral reconstruction (Figure).

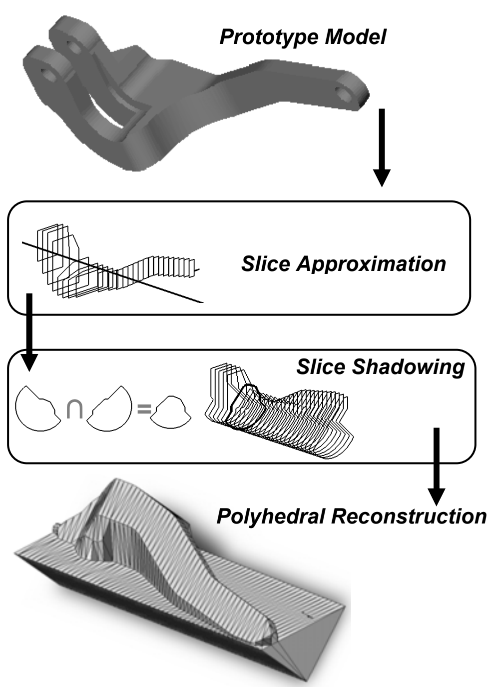


Figure 4 – Method for remaining stock computation, composed of three steps

Slice Approximation

Slice approximation is a pre-processing step used to convert a surface model of the part geometry to cross sectional polygonal data. Since we assume all setups occur about a single theoretical axis, the problem can be represented as a set of cross sections perpendicular to

and along that axis. This provides the distinct advantage of reducing the overall problem to independent analyses of two-dimensional data from a three dimensional model.
However, the preprocessing involves more than simply slicing the model. Subsequent algorithms in the method use a visibility based shadow to approximate remaining stock, so the slices must account for the difference between *visible* and *accessible* regions. If a visibility approach is used on the exact cross section of the part, Type A error will occur for all but the simplest geometries (remaining stock will be underestimated). To avoid this, the cross sectional data must exclude small concave features (i.e. holes and slots) that a tool of specific finite diameter may not be able to access. The approach to counter visibility underestimation is twofold: cross sections are (1) reduced to their own convex hull polygons and (2) redefined as the union of all slices within one tool radius along the axis of rotation. The first measure has the effect of closing concave geometry that is radial relative to the axis of rotation, while the second measure closes concave geometry along the axis. As the final step of slice approximation, the resulting cross sections can be offset in order to approximately account for scallop effects. Since specific tool instances are assumed unknown, the location of scallops is likewise unknown. To this point, offsetting the polygonal cross sections allows compensation for possible scallops and for any amount of intentional under-cutting (e.g. leaving some set amount of material for finishing operations). The net effect of the slice approximation steps described above is shown in Figure 5 with a reconstructed model. In particular, note that the small holes and slots inaccessible by a roughing tool have been eliminated; however, thick features have been slightly over approximated (Type B error). This reduction in accuracy is the cost of using a robust form of visibility analysis during slice alteration; however, it provides a critical degree of safety to ensure tool collisions will not occur.

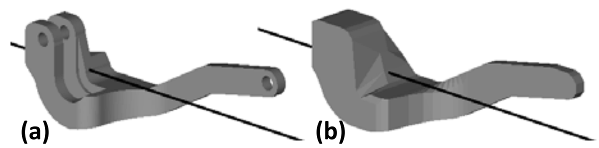


Figure 5 – Example component along axis of rotation, (a) original model and (b) reconstruction of approximating slices showing elimination of inaccessible holes and slots

The methodology begins with an STL model of a part and fixturing geometry (representing the space set in which no machining should take place) is sliced into sets of contours using existing slicing methods. For multiple setup processes, the cross sectional slice planes are set normal to the axis of rotation. For this work, we will define $\mathbb{C} : \{C_1, C_2, \ldots, C_n\}$ as the component data set $\mathbb{C}$ of n slices, each denoted C_i, where l is the total length of component along rotational axis, d is the distance between slices and r is the tool radius. Since this methodology is later used to drive tool path planning, it must <u>never</u> have *Type A* error so that collisions do not occur; equivalently, the amount of remaining stock must never be underestimated. To maintain a lightweight model and avoid the computational complexity associated with collision or machinability algorithms, two steps are taken to further process the slices and avoid underestimating material removal in visible regions that a tool could not actually reach. First, each cross section is reduced to its own convex hull ($C_i = ConvexHull(C_i)$; $i=1\ldots n$) as shown in Figure 6. This step has the effect of eliminating slots or other internal geometry radial with respect to the axis of rotation (any given convex hull is completely visible and accessible by any convex 2D tool silhouette). Second, each slice is redefined as the union of itself and all slices within one tool radii; ($C_i = (C_{i-(r/d)})$ U $(\ldots)$ U (C_i) U $(\ldots)$ U $(C_{i+(r/d)})$; $i=1\ldots n$), as shown in Figure 7.

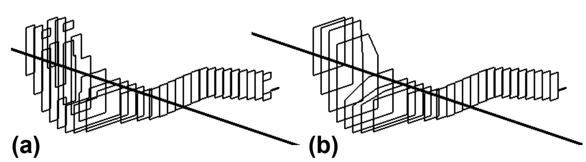


Figure 6 – Each slice (a) is represented as (b) its convex hull polygon

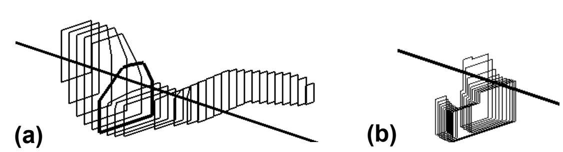


Figure 7 – Each slice in (a) is redefined as the union of slices within one tool radius of the slice (as shown in image b for the bold slice in image a)

This second step offsets all possible collision surfaces by one tool radius and closes small gaps and slots along the axis of rotation. Both of these steps contribute Type B error, where the remaining stock is over-estimated, though not detrimentally so as discussed in the implementation section. In addition to the part geometry, slices must also be created to represent initial stock geometry. Slices can be generated automatically to represent homogenous stock material (bar or prismatic stock), or a model of the beginning stock can be sliced to create cross-sections. Regardless, the beginning stock is assumed to be of convex cross section along the axis of rotation, where we define: $\mathbb{S} : \{S_1, S_2, \ldots, S_n\}$, the stock data set $\mathbb{S}$ of n slices, each denoted S_i. It should be noted that for stock material that is homogenous along the axis of rotation (i.e. bar or round stock), all S_i in $\mathbb{S}$ are equivalent.

Slice Shadowing

Slice shadowing, the second of three steps, is central to the remaining stock analysis. After the model has been represented as a set of cross sections, each cross section is independently modified with a unique 'limited visibility shadow'. Unlike infinite visibility from a given orientation, we explicitly define a finite visibility range to emulate the finite cutting depth common to subtractive processes such as machining. This shadow is readily defined by Boolean operations between the part cross sections and initial stock cross sections. The algorithm is generalized to allow multiple 'cuts' to different depths from different orientations.

Once the component has been approximated as a series of slices, $\mathbb{C}$, the effect of cutting is simulated on each slice C_i independently. For this simulation, we note that the remaining stock will be equivalent to the union of the stock material (cut to a certain depth), the component, and the visibility shadow of the component (where the tool could not reach). Recall that measures were taken during 'Slice Approximation' to account for the difference between visibility and accessibility analysis. For the purposes of detailed discussion, we further define m as the number of cutting orientations, $\mathbb{a} : \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$, the set of cutting angles, $\mathbb{d} : \{\delta_1, \delta_2, \ldots, \delta_m\}$, the set of cutting depths, relative to axis of rotation viewed from angles $\mathbb{a}$, $\mathbb{A}_j : \{A_{1,j}, A_{2,j}, \ldots, A_{n,j}\}$, the set of n intermediate slices, representing the visibility shadow from angle α_j, $\mathbb{B}_j : \{B_{1,j}, B_{2,j}, \ldots, B_{n,j}\}$, the set of n intermediate slices, representing stock material (without component or shadow) from angle α_j to depth δ_j, $\mathbb{R}_j : \{R_{1,j}, R_{2,j}, \ldots, R_{n,j}\}$, the set of n slices representing effect of a single cut from angle α_j to depth δ_j and finally $\mathbb{ReSt}_j : \{ReSt_{1,j}, ReSt_{2,j}, \ldots, ReSt_{n,j}\}$, the set of n slices, representing effect of cumulative cuts from 1 … j. With the final goal of achieving slice set $\mathbb{ReSt}_j$, sets $\mathbb{A}_j$, $\mathbb{B}_j$, and $\mathbb{R}_j$ are computed. Each is presented below:

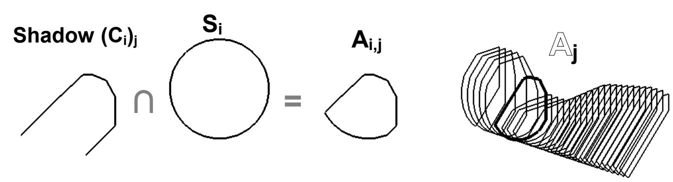


Figure 8 - $\mathbb{A}_j$ is the set of slices $A_{i,j}$ representing a visibility shadow of C_i at angle α_j trimmed to stock cross-section S_i

Slice set $\mathbb{A}_j$ (Figure 8) represents a visibility shadow from angle α_j cast onto set $\mathbb{C}$. Because each C_i is a known polygon, a visibility shadow can be easily determined by finding points in C_i tangent to α_j. To find these points, some vector in the direction of α_j is determined. Then, the cross product between this vector and each segment (span between polygon vertices p_k and p_{k+1}) in the cross section is found; if the sign of this cross product changes between two segments, their shared endpoint is tangent to α_j. In the case presented, the polygonal cross section is known to be convex, thus, there are exactly two points tangent to any given direction vector. Two new vertices, $p_{u,j}$ and $p_{v,j}$, can be added to the polygon point set at some arbitrarily large distance in the direction of α_j from tangency points p_u and p_v). Treating the polygon vertices as a point cloud and recomputing the convex hull yields a 'shadow' contour to some finite distance (the general case of non-convex polygons would require a simple modification to detect the intersection of the shadow ray and other polygon segments). Finally, this 'shadow' is trimmed to the stock cross section via intersection and the result is stored as $A_{i,j}$.

Next, slice set $\mathbb{B}_j$ (Figure 9) represents the effect on stock material of a cut from angle α_j to depth δ_j, assuming for now that no component geometry restricts tool access (we are only considering stock cutting for $\mathbb{B}_j$). First, an arbitrarily large polygon is created to represent the half-space not accessible by a tool at depth δ_j at angle α_j. In practice, this is done by forming a sufficiently large square centered on the origin, translating this square polygon so that the top segment is at depth δ_j, and rotating the polygon by angle α_j about the origin. Then, each $B_{i,j}$ can be computed by trimming, via intersection, the stock cross section (S_i) to the appropriate 'half-space' polygon. In the special case that the stock is uniform along the slicing (cross sectional) axis, all $B_{i,j}$ are equivalent for each cut j and only one $B_{i,j}$ need be computed.

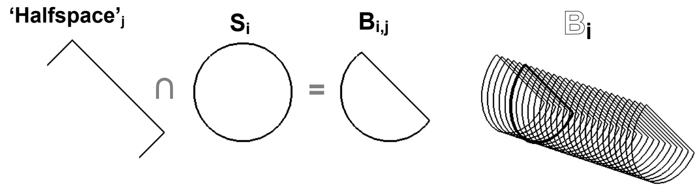


Figure 9 - $\mathbb{B}_j$ is the set of slices $B_{i,j}$ composed of stock material unreachable from angle α_j to depth δ_j, computed by trimming the unreachable 'halfspace' to the stock cross-section

The union of respective slices in $\mathbb{A}_j$ and $\mathbb{B}_j$ yields $\mathbb{R}_j$,(Figure 10) representing the net effect of a single cut from angle α_j to depth δ_j. Without formal proof, we assert that this represents the remaining stock assuming that a tool path is used that has no gouge (removing too much material) or undercut (failing to remove all desired material) motion.

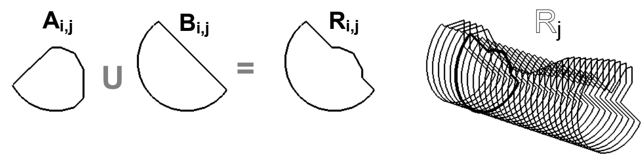


Figure 10 – Single-cut effect is represented by slice set $\mathbb{R}_j$, where each $R_{i,j}$ is computed by joining shadow $A_{i,j}$ and stock material $B_{i,j}$

Finally, to evaluate the cumulative effect of cutting operations, the set $\mathbb{ReSt}_j$ is computed. In the natural sense, one would use each $\mathbb{R}_j$ as the beginning stock material for computing the effect of cut *j+1*. However, this makes computations dependent on each other; it would be more appropriate to compute each $\mathbb{ReSt}_j$ in terms of the preceding $\mathbb{R}_1$ … $\mathbb{R}_j$ such that all $\mathbb{R}_j$ can be computed independently for large scale simulations. Thus, the cumulative effect of cuts ($\mathbb{ReSt}_j$) , (Figure 11) is computed as the cumulative intersection of single-cut effects ($\mathbb{R}_j$), allowing independent calculation of single-cut effects regardless of operation order.

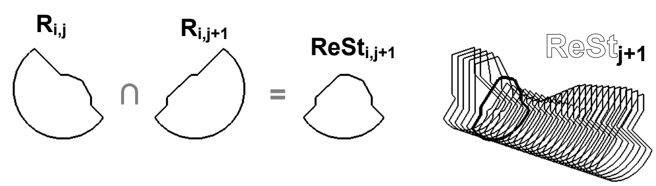


Figure 11 – Each $ReSt_{i,j}$ in $\mathbb{ReSt}_{j+1}$ is computed as the cumulative intersection of preceding single-cut effects $R_{i,j}$, allowing independent computation of $R_{i,j}$

Polyhedral Reconstruction

The resulting contours from the shadowing operation compose a rich data set, though not portable in any widely accepted format. Hence, the third and final step is a polyhedral reconstruction of the stock geometry from the modified slice data, which converts the data to a common format (STL model) for graphics rendering and use in CAD/CAM packages. Polyhedral reconstruction from polygonal cross sections is a relatively well known problem, and in this work we created our own methods unique to this application. Our approach addressed the three primary problems of reconstruction: *correspondence*, *branching*, and *tiling* [1]. However, for brevity we will not include a description of these methods. The primary focus of this paper was the remaining stock calculations, enabled by slice approximation and slice shadowing methods described above.

Implementation

The algorithms presented above have been implemented in software and utilized in the laboratory for four-axis subtractive rapid prototyping (CNC-RP). These algorithms are ideally suited for this system because they provide a robust, reliable estimation of remaining stock. In the CNC-RP process, it is more important to achieve safe and effective results for first-pass success in low volume applications than it is to optimize tool paths at a high level (the expectation is that only one, or very few parts are desired). This remaining stock calculation serves exactly that purpose: while the stock is overestimated in certain situations, it is done so to ensure the integrity of results and eliminate collisions between the tool and underestimated stock material while the machine is allowed to run unattended. To illustrate the results typical in the CNC-RP implementation, reconstructed models for a single component machined in seven setups are illustrated in Figure 1.

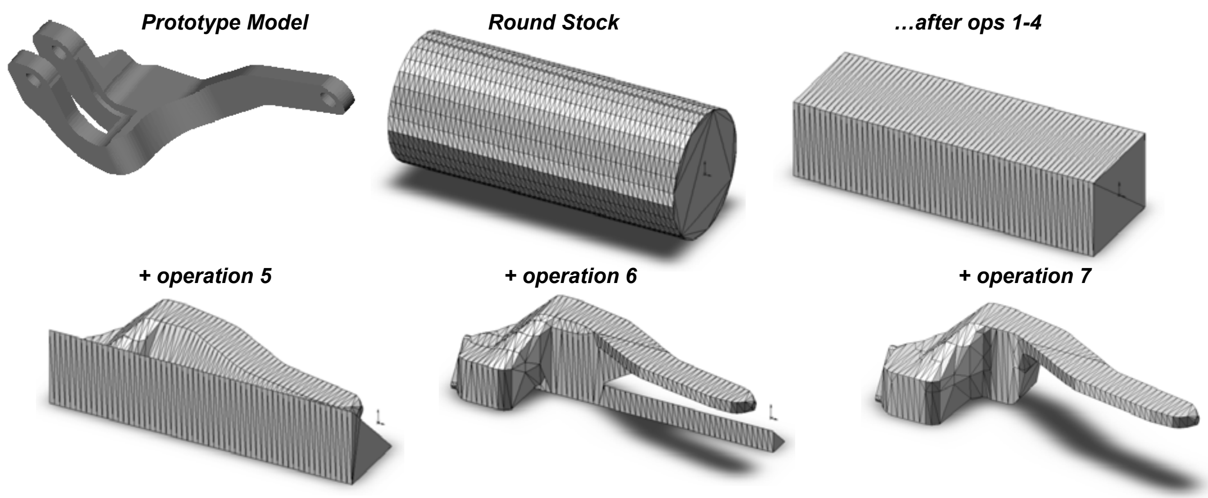
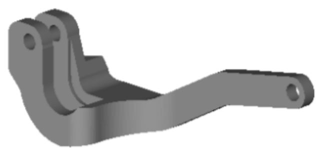
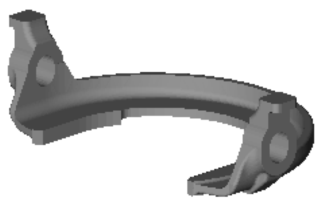
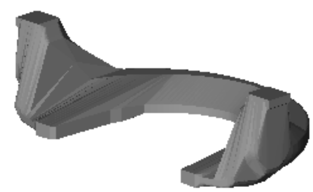


Figure 12 – Remaining stock approximations for the example component from seven setup orientations

The efficiency gains from this implementation have been very significant. While a variety of factors may influence tool path inefficiency, one of the largest contributors is adopting a naïve and overly conservative remaining stock assumption. Overall tool path length has been observed to decrease as much as 65% when utilizing this new remaining stock analysis during tool path generation.

Improved efficiencies are presented for two sample parts (Table 1). Each was assumed to be machined from three inch diameter bar stock with sacrificial supports (consistent with the CNC-RP implementation) using a 0.5 inch flat end mill, 0.04 inch step down, and 75% step over. In laboratory testing, the tool was previously observed performing redundant 'air' cutting, but with the remaining stock implementation it remains engaged in the material during nearly the entire roughing routine. Numerous laboratory tests on a variety of geometries have not revealed any cases of *Type A* error (failure mode of collision with under-estimated remaining stock).

Table 1 – Results (coupling with multi-axis tool path generation) for two example geometries

CAD Model	Reconstructed Remaining Stock Model	Total Setups	Tool Path Length Reduction	Computation Time (3 GHz CPU)
		9	53%	640 milliseconds
		8	64%	640 milliseconds

Limitations and Future Work

This work is limited primarily by the assumption that setups must be about a single axis. However, in practice, it is observed that most geometries can be achieved in this manner (from machinery components to free-form biologic shapes such as human bones). This fact may be due to the motivation behind design for manufacturing to reduce setups for industrial components. Using visibility software in the lab, the authors have found few bones of the human anatomy (i.e. the entire skull and pelvis) that cannot be manufacturing about one axis of rotation. The steps taken during slice preprocessing to avoid underestimating the remaining material (approximation as a convex hull) cause slight overestimation in many cases. This constraint could be relaxed if a machinability algorithm was incorporated and a more robust correspondence algorithm implemented for polyhedral reconstruction. While this would result in a more exact approximation, the machinability algorithm would substantially increase computation time. As a final limitation, the method presented does not convey scallop effects or surface finish in the remaining stock estimation. However, the method presented is not intended to replace conventional verification and simulation techniques for finishing operations; rather, its strength is in advanced, efficient process planning.

Jerard [12] proposes a framework for automated NC code generation via an iterative process between setup orientations, setup sequencing, tool path generation, and tool path verification. With the algorithms we present, setup orientation and sequencing can be improved *without* iteration between computationally expensive tool path generation and verification algorithms. The robust and simple nature of this algorithm lends itself to future simulation or optimization techniques for determining optimal setup strategies. The Z-buffer method was the fastest pre-existing simulation method [3]; Huang and Oliver [11] cite a computation speed of 68.8 instances / second for a three-axis roughing routine (analogous to the operations analyzed here). Even at this rate, the tool path for the example part would have taken several minutes to simulate (for 9069 linear inches of feed move in the example component's NC program), even at modern computing speeds. However, by evaluating the net effect of the tool path rather than each individual move, our method requires only 0.6 seconds to simulate nine different operations from different setup orientations (considering the example part on a 3 GHz CPU). Because the net effect of a tool path requires only milliseconds to calculate, this algorithm is appropriate for application to optimization techniques.

One such application for focus of future work is for the detection of thin material conditions. If operations are not sequenced optimally, machining from opposing setup angles (at or near 180°) can form thin webs or thin strings (Figure 13). These conditions can be readily detected by our algorithm at the slice level.

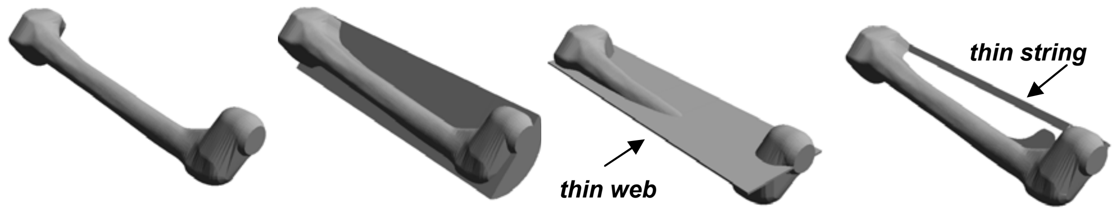


Figure 13 – Examples of thin webs and strings emerging from the multi sided machining of a human femur from round stock

The method presented can also be applied to design analysis. The shadow method can be used to augment visibility algorithms. While visibility and accessibility algorithms investigate the invisible / inaccessible surface area of a component, our new method can be used to evaluate inaccessible *volume* from a given set of angles. One possible application would be evaluating the amount of material that must be added to regions of a component to make molding or forging possible from a minimal number of molds. Another future application is to simulate wire EDM processes since, in fact, wire EDM cannot readily create concave geometry and it is very nearly a line of sight accessible process (for small wire diameters). Thus, the assumptions made for the current algorithm would map directly to a wire EDM process.

Conclusions

This work presented a new method for gross analysis of subtractive processes in multiple setup environments. As opposed to existing methods, focusing on single setup (albeit multi-axis) surface finishing verification, our method provides a robust and efficient means of evaluating subtractive processes. Specific results of interest are a new limited visibility shadowing algorithm and efficient analysis of subtractive processes independent of specific tool paths. The limited visibility shadowing algorithm can be applied to a variety of other situations. Other visibility analyses evaluate surface visibility, and a set cover problem can be formulated to solve for optimal setup angles. Using this algorithm, the problem could be formulated differently to achieve the maximum *accessible volume* in a set cover problem rather than maximum accessible surface area. Moreover, the extremely fast nature of this algorithm lends itself to iterative optimization approaches; which has significant implications to improved process planning. If an objective function can be formulated that uses metrics to identify advantageous or detrimental characteristics of process progress, the millisecond time scale of our algorithms will allow many iterations through setups to identify (at least locally) optimal setup sequences. This could be specifically advantageous in 4-axis SRP, if the desired prototype has complex geometry that will require numerous arbitrary setup orientations about the axis of rotation.

Acknowledgment

Financial support was provided by grants from the National Institutes of Health (AR48939 and AR55533) and Deere and Company (Account 400-60-41).

References

[1] Bajaj, C., E. Coyle, and K.-N. Lin. "Arbitrary Topology Shape Reconstruction from Planar Cross Sections", *Graphical Models and Image Processing*, v58 n6 November 1996.

[2] Blackmore, D., M. C. Leu, and L. P. Wang. "The sweep-envelope differential equation algorithm and its application to NC machining verification", *Computer-Aided Design*, v29 n9 1997.

[3] Bohez, E., N. Minh, B. Kiatsrithanakorn, P. Natasukon, H. Ruei-Yun, L. Son. “The stencil buffer sweep plane algorithm for 5-axis CNC tool path verification”, *Computer-Aided Design*, v35, n12, October 2003.

[4] Boissonnat, J.-D. “Shape Reconstruction from planar cross-sections”, *Computer Vision, Graphics, and Image Processing*, v44 n1 October 1988.

[5] Chappel, L. “The use of vector to simulate material removed by numerically controlled milling”, *Computer-Aided Design*, v15 n3 1983.

[6] Christiansen, H. and T. Sederberg. “Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics”, *Computer Graphics*, v12 n3 1978.

[7] Ekoule, A., F. Peyrin, and C. Odet. “A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours”,

[8] Frank, M., R. Wysk, S. Joshi. “Determining setup orientations from the visibility of slice geometry for rapid computer numerically controlled machining”, *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, v128 n1 February 2006.

[9] Fuchs, H., Kedem, Z., Uselton, S. “Optimal Surface Reconstruction from Planar Contours”, *Communications of the ACM*, v20 n10 October 1977.

[10] Ganapathy, S. and T. Dennehy. “A New General Tiangulation Method for Planar Contours”, *Computer Graphics*, v16 n3 July 1982.

[11] Huang, Y. and J. Oliver. “NC milling error assessment and tool path correction”, *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1994.

[12] Jerard, R., R. Drysdale, K. Hauck, B. Schaudt, and J. Magewick. “Methods for detecting errors in numerically controlled machining of sculptured surfaces”, *IEEE Computer Graphics and Applications*, v9 n1 January 1989.

[13] Jütler, B. and M. G. Wagner. “Computer-Aided Design with Spatial Rational B-spline Motions”, *Journal of Mechanical Design, Transactions of the ASME*, v118 n2 June 1996.

[14] Keppel, E. “Approximating complex surfaces by triangulation of contour lines”, *IBM Journal of Research and Development*, v19 n1 January 1975.

[15] Lauwers, B., J.-P. Kruth, P. Dehonghe, R. Vreys. “Efficient NC-Programming of Multi-Axes Milling Machines through the Integration of Tool Path Generation and NC-Simulation”, *CIRP Annals – Manufacturing Technology*, v49 n1 2000.

[16] Oliver, J. and E. Goodman. “Color graphic verification of N/C milling programs for sculptured surface parts”, *First Symposium on integrated intelligent manufacturing*, New York, ASME 1986.

[17] Pal, P. “Remaining Stock Computation for 3D-Machining in Parametric Regime”, *Journal of Manufacturing Science and Engineering*, v127 n4 November 2005.

[18] Saito, T. and T. Takahashi. “NC Machining with G-buffer Method”, *Computer Graphics*, v25, n4, July 1991.

[19] Van Hook, T. “Real-Time Shaded NC Milling Display”, *Computer Graphics*, v20 n4 August 1986.

[20] Wang, Y. and J. Aggarwal. “Surface Reconstruction and Representation of 3-D Scenes”, *Pattern Recognition*, v19 n3 1986.

[21] Weinert, K., S. Du, P. Damm, M. Stautner. “Swept volume generation for the simulation of machining processes”, *International Journal of Machine Tools & Manufacture*, v44 n6 May 2004.