Operation decomposition for freeform surface features in process planning

Ganping Suna,*, Carlo H. Sequinb, Paul K. Wrighta

aMechanical Engineering Department, University of California at Berkeley, Berkeley, CA 94720, USA
bComputer Science Department, University of California at Berkeley, Berkeley, CA 94720, USA

Received 1 November 1999; revised 15 September 2000; accepted 15 October 2000

Abstract

Machining operations for freeform surface features usually include roughing, finishing for the bottom surface, tapering, and corner rounding. Strategies and algorithms to decompose the overall task into these operations are presented. The decomposition aims at minimizing machining time within the constraints of the specified surface roughness and tolerance, and machine tool safety. The roughing operation can be further decomposed into sub-operations for multiple tools. There are two strategies to decompose the finishing operation into sub-operations: one is based on multiple tools and another is based on multiple tool path patterns. The approaches to select the optimal decomposition values (tool diameters, surface slopes) that minimize machining time are presented. These algorithms are being integrated into a rapid-prototyping service for web-based machining. Design for manufacturability and maximizing process automation are the key priorities in process planning and operation decomposition. © 2001 Published by Elsevier Science Ltd.

Keywords: Freeform surfaces; NC milling; Decomposing; Form features; Process planning

1. Introduction

Freeform surfaces are often used in the designs of electronic appliances, automobiles, and airplanes. They allow designers describe part geometries with functional and aesthetic goals in a convenient manner. In the past decade, freeform surface design and machining has attracted much attention from both academic researchers and industry practitioners. Recent research work in this field has mostly focused on feature recognition and tool path planning.

Feature-based CAD/CAM [5,8,9,12,16] also aims to reduce the interactions between designers and fabricators and to automate process planning. In feature based milling, a part is described as a starting volume or stock and a set of features, representing the volumes removed by machining. Integration of manufacturing knowledge into the design tool allows to check the manufacturability of each feature at the design stage. Thus feature-based CAD can reduce the complexity of operation decomposition, process planning time, and minimize the number of design/fabricator iterations.

The purpose of tool path planning is to generate gouge-free tool paths for a desired geometry, with a certain tool and given cutting parameters. In order to meet given requirements on surface finish while minimizing machining time, the tool path pattern should cover the surface as uniformly as possible. Recently, much research concerns such algorithms to define optimal tool paths for various tools, part geometries, and machines [1–4,7,13].

Most commercial CAD/CAM systems have the capability to design and machine some kinds of freeform surface features. However, process planning and operation decomposition still are major challenges. Some of the problems encountered include:

- Freeform surface features are often machined in multiple operations: roughing, finishing, tapering, and corner rounding. These features have to be decomposed into more cohesive, lower level operations before they can be processed by the tool path generator.
- Tool and cutting parameter selections are critical to both machining quality and cutting efficiency. A large tool may yield high cutting efficiency but cannot clean up corners in a freeform feature, while a small tool can cover all areas, but at a low cutting efficiency. A combination of tools of different sizes is thus typically used. During process planning, the tool and cutting parameters for each operation must be selected to meet design specifications (roughness and tolerance) and maintain machine tool safety, while minimizing machining time.
- Not all manufacturability problems can be detected at the design stage. Some are related to the chosen tools and
their operational parameters, which are only selected at the process planning stage. For example, designers often design pockets that are narrower and deeper than what can be reached with available tools.

Operation decomposition builds a linkage between the feature recognizer and the tool path planner. Its separate functions include:

- Decomposing a feature into a list of machinable operations (roughing, finishing, tapering, and corner rounding)
- Decomposing the roughing operation into sub-operations for multiple tools
- Decomposing the finishing operation into sub-operations for multiple tools and multiple tool path patterns
- Selecting the optimal decomposition values (tool diameters, surface slopes) that minimize overall machining time

After operation decomposition, a list of individual operations each with their local geometries, chosen tools, and applicable type of tool path patterns are forwarded to the tool path planner. The prime concerns are full automation and (almost) guaranteed manufacturability in order to reduce the interaction between designers and manufacturers.

In this paper, strategies and algorithms to automatically decompose a freeform surface feature into a list of feasible operations are presented. The features considered are 3-axis milling features, accessible from a predefined direction. They include 2.5D extruded pocket geometries which may have a freeform surface.

2. Technical definitions

2.1. Freeform surface features

In the context of 3-axis milling, a freeform surface feature can be represented by ‘3D bottom surfaces’ and ‘2D boundary contours’. The latter may carry some attributes such as: tapered angles (draft angles), open conditions, corner radii, top and bottom rim radii [11]. The 3D freeform surfaces defining the bottom of the pocket can be a single patch, or composed of multiple patches, which must be connected together with at least G0 continuity.

The 2D boundary contours defining each pocket can consist of one outside contour and may have one or more inside contours (islands). In some cases, such as injection molds and sheet metal dies, side walls are often tapered with a draft angle from 1 to 5° and rounded with some radius at the top edges. A designer also can assign radii for the sharp corners in the boundaries and for the top and bottom rim of a pocket. For representation simplicity, these various radii are stored as attributes of the corresponding contour element, not as actual geometry. Some parts of the boundaries may be ‘open’, i.e. not limited by side walls. Tools can go across open boundaries without any obstructions (unconstrained) or with the constraints of the geometry of the adjacent pockets (constrained).

Fig. 1 shows a typical freeform surface feature with a generic ‘bathtub’ like appearance. This general definition of freeform surface features can cover most application cases.

2.2. Manufacturability problems for freeform surface features

By design, all surface points in a 3-axis milling feature are visible from one direction, due to the limitations of the tool size and length, not every point may be reachable by an available tool. Examples include places with small curvature radius (Fig. 2(a)), sharp creases (Fig. 2(b)), deep regions (Fig. 2(c)) in the bottom surface, or small corner radii (Fig. 2(d)), and narrow channels (Fig. 2(e)) in the pocket contour. As a result, some areas of the surfaces cannot be machined due to potential tool gouges and interferences. During operation decompositions, tool selections are constrained by minimal curvature, corner radii, minimal channel widths, or maximal depths. If no proper tool can be found for an intended curvature, that operation is not machinable. Another possible failure comes from the cutting parameters. If no proper parameters can be found to meet the roughness and tolerance requirements, the operation also must be rejected. Any such failures must be returned to the process planner for the selection of another

---

Fig. 1. A typical freeform surface feature.
feature decomposition, or must even be sent to the designer for design modifications.

2.3. Operations for freeform surface features

The overall operation decomposition process for freeform surface features is shown in Fig. 3. Generally, based on feature geometry, a freeform feature can be decomposed into four major operations: roughing, finishing, tapering, and corner rounding. Roughing and finishing operations, which occupy most of the machining time, should be further decomposed into sub-operations for the purpose of high cutting efficiency. For the roughing operation (Section 3), the sliced contours are decomposed for multiple tools. For the finishing operation (Section 4), the bottom surfaces are decomposed in two ways: decomposition for multiple tools (Section 4.1) and for multiple tool path patterns (Section 4.2). The tapering (Section 5.1) and corner rounding (Section 5.2) operations are usually carried out by special tools and need not be decomposed further. Table 1 shows the tool, cutting parameters, cutting geometry, and limitation geometry for individual operations.

3. Operation decomposition for roughing

The purpose of rough cutting is to remove as much material as possible so that the remaining material is close to the final geometry. During the roughing process, the tolerance and surface quality are not the major concerns. For the purpose of high cutting efficiency, relatively aggressive cutting parameters can be employed within the constraint of machine tool safety.

3.1. Contour decomposition roughing with multiple tools

For freeform features, rough cutting is often performed by slicing a feature volume into layers, and cutting each sliced layer in a spiral or zigzag pattern similar to 2.5D feature
machining (Fig. 4(a)). In order to prepare operations with multiple tools, each sliced contour should be decomposed into several regions for the different tools (Fig. 4(b)). This section presents an algorithm to decompose the sliced layers for multiple tools. A map that represents the decomposed regions for the various tool diameters is created. Here we study the case where a slice is removed with one large diameter primary cutting tool followed by a secondary cut with a smaller tool. A strategy is described to select the optimal tool diameter for the primary tool, yielding minimal overall machining time for roughing.

Contour decomposition for multiple tools is performed for all pockets in each sliced layer. For machining efficiency, all operations with the same tool are grouped together to minimize the tool exchange times. The algorithm for the contour decomposition can be described as:

**Step 1** Offset the pocket contour inwardly with a distance equal to the primary tool radius, and then remove the intersections/self-intersections on the offset contour. (Fig. 5(a))

**Step 2** Offset back with the same radius to get the area that will be removed by the primary cutting tool. (Fig. 5(b))

**Step 3** Subtract the primary cutting areas from the original pocket contour to get the secondary cutting areas. (Fig. 5(c))

**Step 4** Extend the secondary cutting areas at the boundaries adjoining with the primary cutting areas (open boundaries) in order to remove spurious material between the two areas. (Fig. 5(d))

If the tool paths for each operation were restricted inside the two decomposed regions as they exist after step 3, undesired material would be left along the boundaries between these two regions. In order to remove undesired material, the contours for the secondary cutting areas is extended at step 4. The boundaries shared by adjoining operations are called 'open boundaries'. Open boundaries also exist at stock boundaries and between adjoining features. Fig. 6 shows a generic algorithm to extend a contour at open boundaries without a collision with closed boundaries. This approach is applicable to all possible configurations of open boundaries.

Another approach to deal with the operation decomposition for a 2D pocket has been presented by Veeramani, and Gau [14]. They employ the Voronoi diagram approach to obtain the offset boundaries. The Voronoi diagram approach is only practical for contours consisting of line and arc segments. However, the sliced contours in roughing operation for freeform features often contain spline segments resulting from the intersection of the bottom freeform surfaces and the slicing planes.
3.2. Decomposition map and optimal tool selection

The decomposed regions for a set of primary cutting tool diameters create a decomposition map for multiple tool diameters (Fig. 7). This map is used to find the optimal tool that minimizes overall machining time.

The radius of the secondary cutting tool can be taken as the minimum of inside curvature radii, channel widths, and corner radii of the sliced contours. However, selection of the primary cutting tool is not an easy optimization process. The larger the tool, the higher the cutting efficiency of the primary cutting, but the more area is left for the secondary cutting at lower efficiency. So, in order to maximize the overall cutting efficiency, the areas for these two regions need to be balanced. Based on the decomposition map for multiple tools, the overall machining time for each decomposition can be calculated with Eq. (1), and the one with the minimal overall machining time can be obtained (Fig. 7). Due to the limited number of available tool diameters, the calculation is not complex

\[
T_{\text{cut}} = \sum_{i=1}^{N_{\text{tool}}} \sum_{j=1}^{N_{\text{slice}}} \frac{\text{Area}_{i,j}}{\text{MMRR}_i}
\]

in which, \(T_{\text{cut}}\) is the total cutting time, \(N_{\text{tool}}\) is the number of tools employed in the roughing, \(N_{\text{slice}}\) is the number of the

---

Fig. 5. Algorithm for contour decomposition for multiple tools. (a) Offset contour by the primary tool radius; (b) offset back to obtain the primary cutting areas; (c) subtract to obtain the secondary cutting areas; (d) extend the secondary cutting areas at the open boundaries.

Fig. 6. The process to extend a 2D contour with open boundaries. (a) A 2.5D pocket with open boundaries; (b) the pocket contour; (c) the protection contour for the closed boundaries; (d) the extension contour for the open boundaries; (e) the potential tool location contour; (f) the final tool location contour; (g) the extended contour; (h) the final tool paths.

Fig. 7. Decomposition map for multiple tools in a 2.5D pocket. (a) Removable regions for the nine different tool diameters; (b) comparison of machining times for different tool combinations.
sliced layers, $Area_{ij}$ is the area for the tool $i$ on the layer $j$, and MMRR$_i$ is the Maximal Material Removal Rate for the tool $i$ in roughing for the chosen slice thickness.

Fig. 8 shows an example of operation decomposition with multiple tools for roughing (three tools). The feature bottom contains two convex surfaces and two concave surfaces (Fig. 8(a)). The feature boundaries are fully open. During roughing operation decomposition, the sliced contours (Fig. 8(b)) are decomposed for three different tools (Fig. 8(c)). Due to the contour extension at open boundaries, the final extended contours at each layer for three sub-operations are partially overlapped (Fig. 8(d)).

4. Operation decomposition for finishing

In order to achieve high cutting efficiency as well as high surface quality, we use a combination of two strategies to decompose finishing operations for the bottom surface: (1) decomposition for multiple tools and (2) decomposition for multiple tool path patterns. Fig. 9 give us an overview of the overall decomposition process. The details of the individual decomposition phases will be described in the following subsections.

4.1. Surface decomposition for finishing with multiple tools

The purpose of finish cutting is to remove the remaining material after rough cutting and meet the design specifications in tolerance and surface finish. During finishing, the cutting force is much smaller than during roughing. Quality control becomes the major concern. Generally, due to its high stiffness and low scallop left on the surfaces with the same stepover, a large tool can easily meet design specifications while maintaining high cutting efficiency. On the other hand, similar to rouging, not all areas on the surfaces can be reached by the large tool. So, in finishing, multiple tools should be selected to cover all areas and to maintain a high overall cutting efficiency.

The tool diameter is constrained by the small curvatures, minimal channels and corner radius both on the feature contour and on the bottom surfaces. In this section, the surface decompositions due to the restrictions from both the feature contour and the bottom surfaces are described. Based on the combinational decomposition map for multiple tools, a strategy to select the optimal tools that minimize overall machining time is presented.

4.1.1. Surface decomposition for multiple tools based on the feature contour restriction

The restrictions from the feature contour on a ball end mill are applied to its center position. However, the resulting bottom surface after milling will be the collection of tool contact points which differ from the tool center by a tool radius in the surface normal direction. In order to decompose the bottom surface for multiple tools based on the feature contour restriction, the constraints on tool centers should be converted to that for tool contact points.

In this section, an algorithm to decompose freeform surfaces for multiple tools based on the restriction from the feature contour is introduced. The constrained tool centers are generated by offsetting the feature contour and projecting it onto an offset surface with one tool radius above the desired bottom surface. The corresponding tool
contact points can be obtained by offsetting this contour back to the bottom surface.

Step 1 Offset the feature contour with a distance equal to the primary tool radius and remove self-intersections. (Fig. 10(a))
Step 2 Offset the bottom surfaces with a distance equal to the primary tool radius. (Fig. 10(b))
Step 3 Project the offset contour onto the offset surfaces in the negative $Z$ direction. (Fig. 10(c))
Step 4 Offset the projected contour back to the bottom surface for defining the area for the primary cutting. (Fig. 10(d))
Step 5 Project the boundaries of the primary cutting areas onto the $XY$ plane to define the 2D boundary contour for the primary cutting. (Fig. 10(e))
Step 6 Repeat steps 1–5 with the secondary tool radius to define the feasible area for the secondary cutting.
Step 7 Subtract the primary cutting area from the feasible secondary cutting area to get the actual secondary cutting area.

A surface decomposition map based on feature contour restriction can be created for different tools. This map give us an overview of the feasible areas for a particular tool.

4.1.2. Surface decomposition for multiple tools based on the bottom surface restriction

During surface machining, the center of the ball end mill must be located on the offset surfaces. But not all locations on the offset surfaces are feasible tool locations. If the offset surfaces intersect each other or self-intersect in some regions, the tool cannot move into these intersection regions without gouging. So, by finding the intersection and self-intersection curves on the offset surfaces, the decomposition regions for the various tools can be defined.

- Algorithm

The algorithm for the surface decomposition for multiple tools based on the bottom surface restriction can be described as follows:

Step 1 Offset the bottom surfaces with a distance equal to the primary tool radius. (Fig. 11(a))
Step 2 Find all intersection and self-intersection curves on the offset surfaces. (Fig. 11(b))
Step 3 Offset back the intersection and self-intersection curves to find the decomposition curves on the surface (Fig. 11(b)). The decomposition curves outline the reachable domains for the various tools and separate areas for primary and secondary cutting. (Fig. 11(c) and (d))
Step 4 Project the decomposition curves onto the $XY$ plane to obtain the 2D contours for the secondary cutting areas.

Step 5 Subtract the secondary cutting area contours from the pocket contour to get the 2D contour for the primary cutting areas.

In principle, steps 1–5 could be repeated for the next smaller tool. If we restrict ourself to only two tools, then we have no choice but to mill out all remaining areas with the secondary tool, even if there are creases that cannot be reached completely.

- Decomposition map for multiple tools

A surface decomposition map thus can be created for the different tool diameters. The map gives us an overview of what areas on the surface can be reached with a particular tool (Fig. 12).

4.1.3. Optimal tool selection based on decomposition maps

Tool selection for finishing operations is restricted by small concave curvatures, minimal channels, and corner radii on both the feature contour and the bottom rims. The decomposed regions due to the restrictions from the feature contour and from the bottom surfaces must be merged. The region for the small tool are the union of the results from both decompositions, while the region for the large tool is their intersection.

Based on the merged decomposition map, we can calculate the overall machining time as a function of the primary tool diameters (Eq. (2)) and select the one with the minimal time

$$T_{\text{cut}} = \sum_{i=1}^{N_{\text{tool}}} \frac{\text{Area}_{i}}{\text{MMRR}_{i}}$$

in which, $T_{\text{cut}}$ is the total cutting time, $N_{\text{tool}}$ is the number of tools employed in the finishing, $\text{Area}_{i}$ is the area for the tool $i$, and $\text{MMRR}_{i}$ is the Maximal Material Removal Rate for the tool $i$ in finishing within the constraints of design specifications (roughness and tolerance).

4.2. Surface decomposition for finishing with multiple tool path patterns

Based on the surface slope, the regions on a surface can usually be classified as flat regions, directionally flat regions,
and steep regions. In a flat region, the projection of the surface normal on the Z-axis is larger than the cosine value of the critical slope angle: \( N_z > \cos(A_{\text{slope}}) \). The opposite is true for a steep region. In a directionally flat region, the projection of the surface normal onto a given vector \( V \) in the \( XY \) plane is smaller than the sine value of the critical slope angle: \( N_v < \sin(A_{\text{slope}}) \).

Surface roughness is determined by the density of the tool paths. Ideally, the constant scallop method [2] can minimize machining time for a standard surface roughness. However, it has high computational complexity and limited surface geometries to which it can be applied. We thus use three simpler tool path patterns: the spiral method (Fig. 13(a)), the iso-plane method parallel to the tool axis (Fig. 13(b)), and the iso-plane method perpendicular to the tool axis (Fig. 13(c)). These methods are applied, respectively, to flat regions, directionally flat regions, and steep regions and produce acceptable roughness when used with a constant distance between neighboring tool paths (stepover). During operation decomposition, surfaces are decomposed into flat (or directionally flat) and steep regions based on whether they are flatter or steeper than a chosen critical slope. In the tool path planner, these regions are then covered with corresponding tool path patterns. By combining multiple tool path patterns, a relatively uniform tool path can be obtained at less computation complexity than would be needed for constant scallop tool paths. This combinational method is applicable to any topology for a 3-axis machinable surface.

### 4.2.1. Surface decomposition for flat and steep regions

Given some values for a critical slope, a surface can be decomposed into ‘steep’ regions and ‘flat’ regions, depending on whether the local slope exceeds the critical value or not. The tool path planner then generates spiral tool paths for the flat regions using a projecting method. In detail, the 2D spiral tool paths inside the 2D boundary contours are projected onto the bottom surface and the projected paths are offset perpendicular to the surface by one tool radius. For the steep regions, iso-plane tool paths are generated by slicing the surfaces perpendicular to the tool axis direction and offsetting these isobathic paths by one tool radius along the local surface normal. The actual tool path will no longer be isobathic.

In this section, a novel algorithm to decompose freeform surfaces into flat and steep regions separated by a critical slope value and a strategy to select the optimal critical slope are presented.

- **Algorithm to find iso-slope curves for a single patch**

As shown in Fig. 14, a normal vector \( N(n_x, n_y, n_z) \) at a point on a surface can be decomposed into two components: \( N_{xy}(n_x, n_y, 0) \), the vector projected onto the \( XY \) plane, and \( N_z(0, 0, n_z) \), the vector projected onto the Z axis. The length of \( N_z \) is related to the surface slope angle \( A_{\text{slope}} \) at the point \( P \) by \( |N_z| = \cos(A_{\text{slope}}) \), i.e. when \( A_{\text{slope}} = 0^\circ \), \( |N_z| = 1 \); when \( A_{\text{slope}} = 90^\circ \), \( |N_z| = 0 \). We now construct a ‘\( N_z \) surface’ by plotting the values \( |N_z| \) over the \((u, v)\) domain of the given surface. On this sloping surface, all points with same Z value have the same slope at the corresponding points on the original surface. For 3-axis machinable features, the range of the \( |N_z| \) value on the bottom surface will vary from 0 to 1. Because the ‘\( N_z \) surface’ is constructed over a \((u, v)\) domain that may be larger than the specified pocket domain, a trimmed surface should be recovered. The recovered surface may contain regions with negative \( N_z \) value. The slope angles for these regions are assigned as negative values. So the range of \( N_z \) may vary from \(-1\) to 1. The \( N_z \)s on a surface can transit from negative values to positive values smoothly.

The iso-slope curves, along which the slope of all points is equal to a critical slope, can be obtained...
by slicing the slope surface and then mapping the intersection curves onto the original surface. These are the main steps of this process:

Step 1 Calculate the slope, |Nz|, for an array of sample points on the surface. (Fig. 15(a)). The accuracy of the final iso-slope curves depends on the density of the sampling points. For surfaces with a high degree of variation, more sample points are necessary to reach the same accuracy than for relatively smooth surfaces. On the other hand, denser sampling points mean longer calculation times. One metric to reflect the degree of surface variation is the number of knots defining the spline. Generally, the designer will have chosen knot densities appropriately to represent the curvatures and undulations of the chosen surface. On the other hand, the diameter of the ball end mill chosen to machine the desired surface also provides a good measure of achievable surface complexity. In an experimental test, we found that using sampling point spacings equal to the chosen tool radius gives a numerical error of less than 0.5° in slope angles along the iso-slope curves. This accuracy is enough for surface decomposition by slope.

Step 2 Create a smooth spline surface by interpolating the slopes (|Nz|) of the sample points over the (u, v) domain. (Fig. 15(b)). This task is performed by the ACIS modeling package[17], which creates an interpolating B-spline surface through the coordinates (u, v, |Nz|) corresponding to the sample points.

Step 3 Create a solid volume above this slope surface by closing it off with a suitable bounding box. (Fig. 15(c))

Step 4 Slice this solid at the height corresponding to the critical slope (height = cos(A_{slope})) to obtain intersection curves in the (u, v) domain that represent points with the same slope in the original surface. (Figs. 14(e) and 15(d))

Step 5 Map these iso-slope curves from the (u, v) domain to the (x, y, z) domain, i.e. onto the surface to be machined. (Fig. 15(f))

- Algorithm to define steep regions and flat regions

In the previous section, an algorithm to define the iso-slope curves on a single patch was described. However, in general, a bottom surface may consist of multiple patches, and may be trimmed by adjoining surfaces and boundaries. In this section, a generic algorithm is presented to define the steep and flat regions for all of these cases.

Iso-slope curves obtained from the previous section must form loops, either by themselves or together with the surface boundaries. These loops enclose the steep regions for a complete surface; if the whole bottom surface were flatter than the critical slope, there would be no intersection at all. As described in Section 2.1, a trimmed surface in a pocket can be represented by a complete surface with trimmed boundary contours. The steep regions for a trimmed surface can be obtained by trimming the complete steep region with
the pocket contours. The areas, outside the steep regions but inside the pocket contours, are the flat regions. For multiple patches, the adjoining flat and steep regions are merged respectively. These are the main steps of this process:

Step 1  Find iso-slope loops for all surfaces. (Fig. 16(a))

Step 2  Project the iso-slope loops onto the XY plane to get the boundary contours for the steep regions over the whole (u, v) domain. (Fig. 16(b))

Step 3  Trim the steep regions against the surface boundary contours if the surface is a trimmed surface, by forming the Boolean intersection between the steep regions and the surface boundary contour. (Fig. 16(c))

Step 4  Merge adjoining trimmed patches, by forming the union of all trimmed steep regions.

Step 5  Subtract the steep regions from the pocket contour to obtain flat regions. (Fig. 16(d))

Step 6  Project 2D contours of the flat and steep regions onto the surface, and correspondingly trim the surface into flat and steep regions.

- Cyclide surfaces

For cyclide surfaces, such as spherical, cylindrical, conical, and toroidal surfaces, the topology of the iso-slope curves in the (u, v) domain may be no longer same as that in the (x, y, z) domain. Two separate points in the (u, v) domain may be connected in the (x, y, z) domain. The sliced curves in the (u, v) domain should be carefully connected to form a properly closed loop in the (x, y, z) domain. (Fig. 17)

- Advantages

In Han’s research work [6], the iso-slope curves are obtained by finding initial points on the iso-slope curves, and tracing these points along the curves. Finding initial points for critical iso-slope curves is not easy, and it is even harder to guarantee that all critical iso-slope curves have been found. We have off-loaded this task to the commercially available ACIS solid modeling package [17] by reformulating it as an intersection operation between a solid and a plane. Once all critical curves have been found, it remains to be determined which regions are flat or steep. During the intersection between the ‘Nz solid’ and the critical slope plane, the areas enclosed in the intersection loop are automatically identified as the steep region and the outside areas are the flat regions. Our approach is applicable to freeform surfaces that are single-value height functions.

Fig. 16. Algorithm to define the flat regions and the steep regions. (a) Iso-slope loops; (b) untrimmed steep regions; (c) steep regions; (d) flat regions.

Fig. 17. Surface decomposition for a torus with a 30° tilt angle. The iso-slope curves on the Nz surface in the (u, v) domain: (a) \( A_{\text{slope}} = 15° \); (b) \( A_{\text{slope}} = 45° \); (c) \( A_{\text{slope}} = 75° \). The iso-slope curves in the (x, y, z) domain: (e) \( A_{\text{slope}} = 15° \); (f) \( A_{\text{slope}} = 45° \); (g) \( A_{\text{slope}} = 75° \).
over the \((x,y)\) domain.

- Decomposition map for flat regions and steep regions

   Based on the previous algorithms, a surface decomposition map is obtained that represents a series of iso-slope curves. The decomposition gives us an overview of possible surface decompositions with different critical slopes. (Fig. 18)

4.2.2. Surface decomposition for directionally flat regions and steep regions

   Given a critical slope angle \(A_{\text{slope}}\) and a vector \(\mathbf{V}\) on the \(x\)-\(y\) plane, the surface can also be decomposed into directionally flat regions and steep regions. In the tool path planner, the tool paths for both flat and directionally flat regions are generated by the iso-plane method. The slicing planes for two regions are different. In the directionally flat region, the slicing planes are parallel to the tool axis and perpendicular to the given vector \(\mathbf{V}\). While in the steep regions, the slicing planes are perpendicular to the tool axis.

   By using \(N_t\) instead of \(N_i\) to define a ‘\(N_t\) surface’, the same algorithm described in the previous section can be employed to define the directionally flat regions and the steep regions on the bottom surfaces. Similarly, a surface decomposition map for different decomposition slopes also can be created (Fig. 19).

4.2.3. Critical slope selection

   The critical slope angle, which distinguishes the flat regions and the steep regions, needs to be determined before operation decomposition. A high critical angle means more flat regions for the projecting method, and denser tool paths are necessary to reach the given roughness requirement. The opposite is true for the steep regions. Thus the critical slope angle affects total machining time. The goal is to find the optimal slope that approximately minimizes machining time.

   Based on the decomposition map, the overall machining time for each decomposition slope can be calculated by Eq. (3). The one with the minimal machining time can be selected as the optimal decomposition slope

\[
T_{\text{cut}} = \left( \frac{\text{Area}_{\text{flat}}}{\cos(A_{\text{slope}})} + \frac{\text{Area}_{\text{steep}}}{\sin(A_{\text{slope}})} \right) \frac{1}{\text{MMRR}} \times \text{Area}_{\text{flat}} + \text{Area}_{\text{steep}} = \text{Area}_{\text{all}}
\]

in which, \(T_{\text{cut}}\) is the total cutting time, \(A_{\text{slope}}\) is the critical slope angle to distinguish the flat (or directionally flat) regions and steep regions, \(\text{Area}_{\text{flat}}\) is the area for the flat (or directionally flat) regions, \(\text{Area}_{\text{steep}}\) is the area for the steep regions, and MMRR is the Maximal Material Removal Rate for the employed tool in finishing within

![Fig. 18. Decomposition map for flat regions and steep regions. (a) Decomposition map for a tours with a 30° tilt angle (top view); (b) the Yinyang surface (oblique view); (c) decomposition map for the yinyang surface (top view).](image)

![Fig. 19. Decomposition map for directionally flat regions and steep regions. (a) Decomposition map for a tours with a 30° tilt angle (top view); (b) the Yinyang surface (oblique view); (c) decomposition map for the yinyang surface (top view).](image)
the constraints of design specifications (roughness and tolerance).

Based on this approach, the optimal decomposition slope for a hemisphere surface can be obtained: $A_{\text{slope}} = 49^\circ$ for flat regions and steep regions, and $A_{\text{slope}} = 41^\circ$ for directionally flat regions and steep regions.

4.3. Combination of the surface decomposition

Two strategies have been shown for decomposing a surface for multiple tools and for different tool path patterns. The final decomposition regions are the intersection of both decompositions (Eq. (4)). The tool and tool path pattern for each region are defined automatically.

$$\text{Region}_{\text{tool}: i} \cap \text{Region}_{\text{method}: j}$$

5. Finishing operations for tapered walls and rounded edges

5.1. Tapering operation

A separate finishing process with a tapered end mill is typically used to produce slightly tapered walls (Fig. 20(a)). In a freeform surface feature, the tapered wall is represented by a 2D boundary contour marked with the attribute of a 1–5° draft angle.

The size of the chosen tool is restricted by the minimal curvature radius and minimal channel on the boundary contour, and the corner radii on both the boundary contour and the bottom rim. The tool length is restricted by the depth of the pocket. Among all applicable tools, the one with the maximal diameter-to-length ratio is selected for the purpose of higher stiffness and less tool deflection. The input geometry for the tool path planner includes the bottom surface and the 2D boundary contour. During the tool path planning, isobathic tool paths are generated by offsetting the tapered wall inwardly with a tool radius and then slicing the offset wall (Fig. 20(b)).

If there is no draft angle but the surface finish requirement on the wall is high, an additional finishing operation with an end mill is also necessary. Its tool selection and geometry preparation process are same as that for the tapering operation.

5.2. Corner rounding operation

In the mold and die design, the designer often wants to round the top rim edges at the upper pocket contours. This can be achieved by an extra path along the top edges with a special milling cutter called a ‘corner rounding mill’ with a concave rounded tip (Fig. 21(a)). In a freeform surface feature, the top rounded rim is represented by a boundary contour carrying the rounding radius as an attributes.

During tool selection, the corner radius of the tool should be exactly same as the rounding radius on the top rim. The tool size is restricted by the minimal curvature radius, minimal channel and corner radius on the boundary contour. The input geometry for the tool path planner is the top rim edge. In the tool path planning, the tool path can be generated by
offsetting the top rim edge inward by the difference between the radii of the chosen tool and rounding corner.

In some cases, the top rim edges may not be planar due to feature interaction from another freeform feature on the top (Fig. 21(b)). In this case, the corner rounding mill cannot be used to cut the top rim, because the envelop of the sweeping tool along a 3D curve cannot match the desired top rim. The top rim should then be machined as a freeform surface by a ball end mill. The surface geometry can be obtained by making a blending operation between the bottom surface of the top feature and the wall of the embedded feature. The tool size is restricted by the minimal curvature radius, by the minimal channel, and by the corner radius on the boundary contour. In the tool path planner, a zigzag tool path is usually generated for the blending surface (Fig. 21(c)).

6. An application example

Fig. 22 is an example that shows the application of the operation decomposition algorithms discussed. The part is resulted from a conceptual design exercise, aiming at making a 3D surface corresponding to the well-known 2D Yinyang symbol dividing a circle into two complementary halves. The solution chosen is composed entirely from pieces of cyclides, i.e. spheres and tori in particular. The top face of the Yinyang part includes convex and concave features with smooth transitions between them. It consists of two toroidal patches and two spherical patches joined with G1 continuity. The bottom of the whole part is a hemisphere with 61 mm (2.4") diameter. Two freeform surface features are recognized in this part: one forming the whole top face, and the other the bottom hemisphere. Each feature is decomposed into three operations: one for roughing and two for finishing. The critical decomposition slopes are 60° for the top face and 45° for the bottom face. For the roughing operation, the contours for the roughing operation

![Fig. 22. The Yinyang parts.](image)

![Fig. 23. The operation decomposition for the Yinyang part.](image)
are extended at the open boundaries, and spiral tool paths are applied to these extended contours in the tool path planning. For the finishing operation, the bottom surfaces are decomposed into flat regions and steep regions. Spiral and iso-plane tool paths are generated for the flat and steep regions, respectively, in the tool path planning. Fig. 23 shows the whole process.

7. Conclusion

Machining operations for freeform surface features include roughing, finishing for the bottom surface, tapering for the wall, and corner rounding for the top rim. In this paper, strategies and algorithms to decompose the overall task into these operations and to further decompose the latter into sub-operations are presented. The decomposition aims at minimizing machining time within the constraints of the specified surface roughness and tolerance, and machine tool safety. These algorithms are being integrated into CyberCut [10,11,15] service for web-based machining at U. C. Berkeley. Design for manufacturability and fully automated process planning are the prime concerns in our environment, and these concerns are also reflected in the operation decomposition process.

There are two main limitations in our current work: (1) In the optimal tool selection based on the decomposition maps, the complexity of exhaustive searching exponentially increases with the number of tools. In practice, a reasonable way is to limit the number of selected tools to 2 or 3, but the selected set of tools may not be the global optimal point though not far from it; (2) In the finishing operation decomposition for multiple tools, the surface offset and self-intersection detection algorithms usually is time-consuming. These problems will be our future research focuses.1

Acknowledgements

CyberCut is a joint research project between Mechanical Engineering and Computer Science at U. C. Berkeley. The project is sponsored by the National Science Foundation.

References


1 At the time of publication, these two problems have been solved. Interested readers can send email to the authors.
Ganping Sun is a postdoc researcher in the mechanical engineering department, University of California at Berkeley. He received his PhD in Mechanical Engineering and MSc in Computer Science from UC Berkeley in 2000, and MSc in Mechanical Engineering from Tsinghua University in 1993. His research interests include Internet based CAD/CAM, process planning, manufacturing process simulation, geometry modeling, and rapid prototyping, in particular for freeform surfaces.

Paul K. Wright’s research work and teaching have focussed on manufacturing processing operations, robotics, expert systems, open-architecture control, rapid prototyping, Internet-based CAD/CAM, the management of technology and distance learning. He has written three books: 21st Century Manufacturing, Metal Cutting (with Edward Trent), and Manufacturing Intelligence (with Dave Bourne). He was born in London, educated at Birmingham University and has held positions at Cambridge University, Auckland University in New Zealand, Carnegie Mellon University in Pittsburgh, and the Courant Institute of Mathematical Sciences in New York. He is presently the A. Martin Berlin Professor of Mechanical Engineering, the Co-Chair of the Management of Technology Program, and the Associate Dean of Distance Learning at The University of California, Berkeley.

Carlo H. Squin is a professor of Computer Science at the University of California, Berkeley. He joined the EECS faculty in 1977 after having worked for 6 years at Bell Telephone Laboratories on the design and investigation of Charge-Coupled Devices. In the early 1980s, he introduced, jointly with D. Patterson, the ‘RICS’ concept to the world of microcomputers. He was head of the Computer Science Division from 1980 until 1983. Subsequently he concentrated on Computer Aided Design tools for the design of integrated circuits. His research interests now lie primarily in the fields of Computer Graphics and CAD tools for 3D geometry, architecture, and mechanical engineers. He is particularly interested in the issues that transcend individual tools, such as consistent and efficient data representations and good user interfaces.