NURBS-based adaptive slicing for efficient rapid prototyping

Weiyin Ma*, Wing-Chung But, Peiren He

Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China

Received 3 July 2003; received in revised form 27 January 2004; accepted 2 February 2004

Abstract

This paper presents slicing algorithms for efficient model prototyping. The algorithms directly operate upon a non-uniform rational B-spline surface model. An adaptive slicing algorithm is developed to obtain an accurate and smooth part surface. A selective hatching strategy is employed to further reduce the build time by solidifying the kernel regions of a part with the maximum allowable thick layers while solidifying the skin areas with adaptive thin layers to obtain the required surface accuracy. In addition, it provides a generalization to the containment problem with mixed tolerances for slicing a part. The article also developed a direct method for computing skin contours for all tolerance requirements. Some case studies are presented to illustrate the developed algorithms and the selective hatching and adaptive slicing strategy. The developed algorithms have been implemented and tested on a fused deposition modeling rapid prototyping machine. Both the implementation and test results are discussed in the paper.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Direct and adaptive slicing; Selective hatching; Rapid prototyping; Fused deposition modeling

1. Introduction

Rapid prototyping (RP) often refers to a class of layer-based manufacturing technologies that first appeared for industry use in about 15 years ago. In contrary to traditional material removal processes, these techniques build a part by gradually adding materials layer by layer [1,2]. The processes are fully automatic. They offer a number of competitive advantages over traditional manufacturing processes and are particularly useful for rapid product development. Commonly used RP processes include stereolithography, selective laser sintering (SLS), fused deposition modeling (FDM), laminated object manufacturing and 3D printing. Several issues in RP, such as new processes, material properties, part surface quality, build time, applications and tooling are active research topics. In the area for surface accuracy improvement and build time reduction, there have also been researches during the past decade on the optimization of build orientation and adaptive slicing.

As a part is built up layer by layer with RP processes, a staircase effect is a major factor affecting the surface smoothness. Its influence is determined by the layer thickness and the local part geometry. If the layer thickness is computed based on the local geometry, the staircase effect can then be controlled to a user-specified tolerance level. In general, there are two slicing approaches for the determination of the layer thickness, i.e. uniform slicing and adaptive slicing. Uniform slicing is the simplest approach for slicing a part at equal intervals. It is widely used for commercial RP systems in industry today. It is however a compromising issue in determining the layer thickness. If the layer thickness is small, one obtains a smooth part. It may however result in many layers and long build time. On the other hand, if the layer thickness is large, the build time is short, but one may end up with a part having a large staircase effect.

For resolving the above compromising issue, adaptive slicing has been proposed to generate a variable layer thickness based on the local surface geometry and a pre-specified smoothness requirement. In literature, one may find a number of attempts for adaptively slicing a part. One of the approaches was reported by Dolenc and Mäkelä [3] for faceted surfaces. The angle of the surface normal is used to predict a possible selection of the layer thickness.
Sabourin et al. [4] proposed another method by first subdividing the model space into uniform slabs. The thickness of these slabs equals to the maximum acceptable layer thickness of a particular RP process. Each of these slabs is then subdivided into uniform thinner layers such that the produced cusp height is within a given tolerance. A carefully designed adaptive slicing algorithm for STL models can be found in Ref. [5]. Cormier et al. [6] and Mani et al. [7] also used a region-based variable tolerance for adaptive slicing. All these approaches are based on an STL interface, the de facto standard of the RP industry. The STL data is a faceted approximation of an object that can only be used to precisely describe planar surfaces. In case of free-form shape, one often needs to create a very large STL file in order to satisfy the accuracy requirement. A lower precision will create a smaller STL file, but the accuracy of the part surface may not be sufficient. The pros and cons of the STL interface have been widely discussed in the RP community during the past years [8–10].

Instead of going through the STL interface, several researchers developed algorithms for directly slicing a computer aided design (CAD) model to maintain its original accuracy, i.e. a direct slicing approach. Among these approaches, Kulkarni and Dutta [11] developed an algorithm for determining a variable layer thickness for an object represented in parametric form. Normal curvature in the vertical direction was used to determine the maximum allowable layer thickness for the surface at referenced level/points with a pre-specified cusp height. To improve the computational performance, Lee and Choi [12] proposed an approach to use both the skin contours and vertical character lines of the part for the determination of the adaptive layer thickness. Jamieson and Hacker [13], on the other hand, utilized a different algorithm to perform direct and adaptive slicing. The model is sliced using a standard layer thickness and compare the contours of the present layer with those of the previous layer to see if the layer thickness can be increased. If the difference between the contours on two slices is small, the maximum layer thickness is first attempted. Otherwise, a middle slice between the standard slice and the previous slice is created. Once again, the contours on the newly created slice are compared to the previous one. The procedure is repeated until a maximum allowable layer thickness is found. With this approach, one may need to compute many slices in order to find an acceptable layer thickness. In addition, Hope et al. [14] slice an object using layers with sloping boundary surfaces that closely match the part surface. Their slicing procedure can greatly reduce the staircase effect, but may not be suitable to commonly used commercial RP systems. Pandey et al. [15] also proposed to consider the actual edge shape of individual layers when using adaptive slicing for FDM prototyping.

In addition to direct and adaptive slicing, Sabourin et al. [16] also presented an interesting approach for tackling the compromising issue of smooth surface finish and fast build time. An object is at first uniformly sliced to create slabs with maximum allowable layer thickness of the RP system similar to the approach reported in Ref. [4]. Contour offsets are further generated to separate the parts within a slab into exterior and interior regions. The exterior regions are further sliced into uniform thinner layers based on the geometric and tolerance requirements. The exterior regions of a part, therefore, built with thin layers to achieve an acceptable surface finish, while its interior regions are built with fast, thick layers to reduce the overall build time. The slicing procedure is based on STL files. This approach is further extended by Tyberg et al. [17] based on local features.

A number of other issues are also critical for producing accurate prototypes models in a short time. The part build orientation for RP processes is one of such issues affecting the surface quality and the build time. In Ref. [18], Allen and Dutta described a method to determine the best orientation based on the criteria of part stability and minimum area of contact with support structures. Frank and Fadel [19] developed an expert system that considered three parameters affecting the fabrication of the part, i.e. surface finish, build time and necessity of support structures. The system interacts with a user to recommend the best build direction. These parameters were also considered by Lan et al. [20] and Xu et al. [21] as criteria to find the optimal build orientation for stereolithography, one of the most successful commercial RP processes. Xu et al. [22] also addressed the issue on optimal build direction for other RP systems. Hu et al. [23] and Yang et al. [24] investigated the issue for the case of hybrid prototyping incorporating machining processes. Peak features, such as peak points, sharp edges or flat planar surfaces, are important landmarks of an object and must be accurately reproduced for satisfying tolerance requirements or for aesthetic reasons. Several researchers have also addressed this issue [3,11,12].

This article further addresses various issues for the development, implementation and realization of adaptive slicing algorithms to tackle the compromising issue of smooth surface finish and fast building speed and of selective hatching algorithms to further improve the building speed. The complete procedure is subdivided into the following major steps.

- At first, all peak features of the CAD model are identified from the model surfaces as a set of feature points in order to preserve the peak features of the object. These feature points subdivide the CAD model into slabs along the slicing direction, usually the z-direction. Feature points are sorted according to the slicing direction as input for the adaptive slicing algorithm. Preserving peak features for model prototyping have been addressed in Ref. [3]. In this paper, algorithms directly operate upon a CAD surface model are discussed.
An adaptive slicing algorithm based on surface curvature along the vertical direction at the reference level/points is applied to each of the slabs with a pre-specified cusp height tolerance, the minimum and maximum layer thickness. The skin contours on each layer are obtained from the allowable layer thickness, local geometry information and the given tolerance. The adaptive slicing algorithm presented in this paper is developed based upon general discussions reported in [11] and our algorithm directly operates upon non-uniform rational B-spline (NURBS) surfaces, the de facto standard of the CAD industry.

The issue for maintaining desired tolerances is often called a containment problem in layered manufacturing [11] and either a positive or a negative tolerance can be used depending on applications of the prototype model. For maintaining the desired tolerance among layers, Kulkarni and Dutta [11] used a ‘bottom-up’ and a ‘top-down’ approach for determining the skin contours. This paper provides a direct method to compute the skin contours with the computed adaptive layer thickness and required cusp height tolerance. With the developed algorithm of the present paper, the cusp height tolerance can be either positive, negative or their arbitrary combination and hence the algorithm provides a satisfactory solution to the containment problem.

To further shorten the build time, a selective hatching algorithm is developed to selectively solidify the areas on the current layer. Starting from the bottom layer of the part upwards, if the accumulated layer thickness approaches to the maximum layer thickness of a certain RP process, a set of common ‘offset’ contours on these several layers are computed. These common offset contours are computed as the 2D offset contours of the maximum inscribed external surface contours or the minimum envelope of the internal surface contours with a pre-specified wall-thickness. The original contours and the common offset contours define the skin of the part. We can then solidify the skin regions at all layers to ensure an accurate part surface, and fill the internal space within the common offset contours on these several layers in a single step to further reduce the build time.

In addition, all the algorithms presented in this paper have been implemented and tested on a FDM RP machine. Given the same tolerance requirement, the results show that significant build time reduction can be achieved with the developed adaptive slicing and selective hatching algorithms.

The above selective hatching strategy has the same objective as that of an approach reported in Refs. [4,16] for producing an accurate and smooth external part surface with fine layers for skin regions while retaining fast building speed using a large layer thickness for internal regions for reducing the build time. However, both the general procedure and details of the algorithms are different:

- The algorithms reported in this paper directly operates upon a NURBS-based CAD surface model, while the approach of Refs. [4,16] slices a part through the STL interface.
- The slicing algorithm reported in this paper produces global adaptive layers. The approach of Refs. [4,16], on the other hand, subdivides the entire model into slabs according to the maximum permissible layer thickness and uses an uniform layer thickness within individual slabs, though the layer thickness for different slabs may be different.
- It should be noticed that this paper also use the word ‘slabs’, but they are totally different from that used in Refs. [4,16]. In this paper, we subdivide the object according to peak features and we call the individual volumes after subdivision slabs.
- In the present paper, a direct method is used to calculate the skin contours of the next layer based on the layer thickness, tolerance requirement and the exact geometric parameters of the current layer. The skin contours of Refs. [4,16] are computed from the contours of that layer and tolerance requirement.

Details of the algorithms summarized in this section will be elaborated in the following sections. The build orientation is also an important issue in layered manufacturing. It is however beyond the scope of the present paper. It is assumed that the build orientation has been determined and transformed as the Z direction before using the slicing procedure. Shrinkage and laser beam compensation are some other issues that must be properly addressed for many of the layered manufacturing processes. These issues are not discussed in this paper either and one may consult [1,3,8] for further discussions. This paper is mainly focused on the development and implementation of the adaptive slicing and selective hatching algorithms based upon a NURBS surface model. Some of the developed algorithms, such as the adaptive slicing with arbitrary tolerance control and the general algorithm for selective hatching, can also be extended for processing models represented in STL format.

2. Adaptive slicing of a NURBS-based CAD model

The approach discussed in this section can be termed as a direct and adaptive slicing approach. The use of adaptive slicing with variable layer thickness can obtain the minimum number of layers along a given direction that satisfy the cusp height requirement or other tolerance criterion. The build time is thus reduced. At the same time, direct slicing of a CAD model can avoid the problems related to the STL interface and thus improve the slicing accuracy. The following subsections detail the slicing
procedure mentioned in the previous section. While our implementation was based on NURBS geometry [25], the general approach discussed in this and the following sections are also applicable to other geometries.

2.1. Identification of peak features

Peak features of an object are easily missed when performing slicing in layered manufacturing [3]. Fig. 1 shows two slicing strategies. All peak features can be preserved with variable layers shown in Fig. 1(b). This is however not the case for uniform layers shown in Fig. 1(a).

A peak feature may occur at places including a corner, a boundary curve, and an interior area of a surface. A peak feature may also be a horizontal edge or a horizontal face. For the purpose of layered manufacturing, all peak features can be handled with the z-coordinates of character/feature points on the individual surfaces. These character points can be subdivided into three classes, i.e. peak/character points P_c, boundary/curve points P_b, and interior/surface points P_i. Horizontal feature edges or feature faces can be represented by any point on the edge or face and, with the following algorithms, they are marked by one of their end/corner points. Peak features are also called as extreme features in this paper.

With the above notations, all the corner points of a surface are first taken as candidates of peak feature points. They are further examined with tighter criteria. A corner point is considered as a peak feature point only if it is a sharp corner, or it is a smooth corner, but the surface normal direction at the corner position is parallel to the slicing direction, i.e. the z-axis. The latter case automatically covers feature faces as, for any point on such a face, the surface normal is always parallel to the slicing direction. In a similar way, all the extreme points on a boundary curve, if any, are those where the surface reaches either a local minimum or a maximum and can again be solved by minimizing 

\[ F(u, v) = \sqrt{z_u^2 + z_v^2} \]

subject to \( u_{\text{min}} \leq u \leq u_{\text{max}} \) and \( v_{\text{min}} \leq v \leq v_{\text{max}} \), where \( z(u, v) \) is the z-coordinate of the surface in 3D space.

For trimmed surfaces, it is possible that an extreme point found by minimizing \( F(u, v) = \sqrt{z_u^2 + z_v^2} \) might not be on the trimmed area. For this reason, all extreme points \( P_s(u, v) \) are further verified to see whether they are lying on the trimmed surface or not. Those extreme points located outside the trimmed surface are rejected. The verification is done in the 2D parametric space. Since the boundaries of a trimmed surface can be defined as 2D curves in the parametric uv-plane, it is possible to determine whether an extreme point should be accepted or rejected by comparing the location of the extreme point to the regions formed by the 2D parametric boundary curves in the uv plane. Fig. 2 shows a spherical surface after trimming and the parameter region enclosed by the boundaries on the uv plane. If the parameter \( (u, v) \) of an extreme point is within the shaded region shown in Fig. 2b, then the extreme point is accepted. One can draw a line in the uv-plane passing through \( (u, v) \) along the direction of the u-axis, and count the number of the intersection points of the line with the boundary curves in the 2D space. If the number of intersection points on one side of \( (u, v) \) is an odd number, the extreme point is then on the trimmed surface.

It should be noted that there might be multiple extreme points on a single curve or surface. The minimization of \( F(t) \) and \( F(u, v) \) should therefore be applied to all curve segments and surface patches, which can be realized by using multiple starting points when solving the minimization problem. In case of a flat surface area, the minimization simply returns one (arbitrary) point on that area as a candidate peak feature point. It is also possible that an extreme point on a boundary curve could be the same as that found on a surface. All extreme points \( P \) on the NURBS surface model should be examined for possible elimination of duplicate points. Also, since only the z-coordinates of peak feature points are

![Fig. 1. Two slicing strategies: (a) peak features lost; (b) peak features preserved.](image)

![Fig. 2. A trimmed spherical surface and its parameter space: (a) the trimmed surface; (b) the parameter region \((u, v)\).](image)
needed for preserving peak features, the \( z \)-coordinates of all peak feature points are then extracted to represent the corresponding peak features. They are finally sorted and repeated \( z \)-coordinates are eliminated from the list. The final extracted unique peak feature points can thus be represented as a union of all identified, i.e. \( P = (\bigcup Z_p) \cup (\bigcup Z_c) \cup (\bigcup Z_s) \). These peak features, represented by a list of \( z \)-coordinates, are finally used to subdivide the object into slabs for further adaptive slicing.

2.2. Layer thickness determination

The tolerance requirement of a RP part is an important input for layer thickness determination. The tolerance distribution can be either negative, positive or a combination of both as shown in Fig. 3. The negative tolerance shown in Fig. 3(a) can be used for situations where the fabricated RP part acts as a core pattern. The positive tolerance shown in Fig. 3(b) is often used in situations where a post-treatment operation, such as polishing, can be entertained. Both cases of negative and positive tolerances are addressed in Ref. [11] as a containment problem. In other general situations, a mixed tolerance as shown in Fig. 3(c) might be desirable for faithfully represent the model shape or for satisfying other fitting conditions. This section provides a generalization of the containment problem and a user may use an arbitrary combination of both negative and positive tolerances. It is therefore possible for users to control the fitting conditions whether the RP model is to be used as a core pattern or followed by any post-treatment operations. In this case, the same distribution of the mixed tolerance occurs at all layers. It is different from other situations mentioned in Refs. [4,11] where the RP part is skewed.

The acceptable layer thickness at a point on a surface is often determined by an allowable cusp height and the local surface shape in the vertical direction. The local surface shape in the vertical direction at a point is represented by an intersection curve between the surface itself and a plane passing through the point. The plane normal is the tangent direction of the contour at that point. For determining the layer thickness, the curve is locally approximated with a circle whose radius is determined by the curvature of the curve at that point, the so-called normal curvature in the vertical direction in Ref. [11]. See Appendix A for further details for determining the adaptive layer thickness.

When the adaptive slicing distance is available, contours of the next layer is computed as the intersection contours between the slicing plane and the NURBS-based surface model. Since the algorithms are implemented in the Unigraphics environment, contour intersection is found by using Unigraphics through its application programming interface [26].

2.3. Computing skin contours for solidification

Depending on tolerance requirement, skin contours might not be the same as the sliced contours as shown in Fig. 4. A top-down and bottom-up approach is used in Ref. [11] for determining the skin contours in case of pure positive or pure negative tolerances. With this approach, the skin contours can be readily determined when the vertical coordinate of the surface normal maintains its sign on any given contour. Otherwise, the upper slice and the lower slice of a layer are projected onto a common horizontal plane to determine an inner and an outer closed contour. The inner and outer contours are then regarded as the skin contours with negative and positive tolerance, respectively.

In the present work, a direct method is proposed for computing the skin contours based on the parameters of the present layer. The approach is similar to the determination of the layer thickness. It computes skin points with layer thickness \( d \) and is applicable to all cases with either negative, positive or mixed tolerance requirements. Since the surface curvature information (the curvature radius \( \rho \)) at the present layer is readily available after computing the layer thickness, the skin contours can be effectively computed based on various cases of Fig. 4. See Appendix B for further details regarding the formulation for skin contour evaluation.

Fig. 3. Tolerance distribution: (a) negative tolerance only; (b) positive tolerance only; and (c) mixed tolerance.
3. Procedures for selective hatching

The use of the variable layer thickness improves the surface quality and shortens the build time for layered manufacturing. The process may however produce very thin layers when slicing objects having curved surfaces and the build time may still be very long. To further improve the building speed without sacrificing the surface accuracy, a selective hatching strategy is incorporated with adaptive slicing. Only the skin regions will be scanned/solidified at
all layers to ensure a smooth part surface. The internal areas will be scanned only when the accumulated layer thickness approaches to the maximum allowable layer thickness of a particular process so as to maintain fast building speed. Based on this idea, we subdivide a layer into skin and internal regions. In other words, the skin regions of a layer are built with the adaptive layer thickness while the internal or the kernel regions are built with the maximum possible layer thickness of a RP process. The idea is shown in Fig. 5.

Let us assume that the adaptive layer thickness at a level $z_i$ is $d_i$. One can find an integer $n$ such that $\sum_{i=0}^{n-1} d_i = D < \sum_{i=0}^{n} d_i$, where $D$ is the maximum layer thickness allowed by a RP system/process. These $n$ layers $d_i$, for $i = 0, 1, \ldots, n - 1$, are then grouped as a hatching unit for determining the skin and the kernel regions. To find the common separation contours between the skin and the kernel regions for all these layers, the skin contours of these $n$ layers are first projected onto a common horizontal plane. The innermost contours of the external skin contours and the outermost contours of the internal skin contours are further identified. The separation contours between the skin region and the kernel region are then the offset contour segments of these extreme skin contours, i.e. the innermost and the outermost skin contours of a hatching unit, with an offset direction pointing towards the material side of the part surface. The offset distance is called the (minimum) wall thickness for the skin region. Fig. 6 shows various contours and the skin and kernel regions. In Figs. 6 and 7 and also for later discussions, the innermost contours of the external contours and the outermost contours of the internal skin contours are simply marked as or called ‘internal contours’ and ‘external contours’, respectively, for simplicity and the partition contours between skin and kernel regions are marked as ‘offset contours’. As shown in Fig. 5, the computed partition contours between the skin and the kernel regions are applied to all layers of the hatching unit.

To determine the skin and kernel regions, a simple three-phase approach as shown in Fig. 7 is adopted. During the first phase, all the offset contours (Fig. 7a), both internal and external, are calculated according to the given wall thickness. During the second phase, internal and external contours are processed independently for the removal of overlapping areas. For internal offset contours, all contour segments falling in any other internal offset contours having intersections are removed and the remaining internal offset contour segments are reorganized (Fig. 7b). For external offset contours, all contour segments falling outside other external offset contours having intersections are removed and the remaining external offset contour segments are reorganized (Fig. 7b). During the third phase, all external contour segments falling inside any other internal contours having intersections and all internal contour segments falling outside any external contours having intersections are removed. The remaining contours...
and contour segments after reorganization are then the partition contours for the skin and kernel regions (Fig. 7c). One may also first determine the skin and kernel regions for each layer in a hatching unit. The kernel regions of the entire hatching unit are then the common areas of the \( n \) layers. Since after adaptive slicing all contours are defined by discrete contour points, both contour offsetting and offset contour processing are built upon contour points/segments.

4. Implementation and interfacing with a FDM machine

The adaptive slicing and selective hatching algorithms reported in this paper have been implemented as two software modules and tested on a FDM RP system [27]. The adaptive slicing algorithm is implemented on a Unigraphics R17 (EDS) environment [26] as an adaptive slicing module (ASM). Given a set of NURBS surfaces and slicing parameters, the ASM module determines a variable layer thickness according to local surface geometry and computes all skin contours for each of the adaptive layers. The output of the ASM is then used as the input for another own-developed software module, i.e. the selective hatching module (SHM), for selective hatching area computation. This module also works collaboratively with the QuickSlice software [27] of Stratasys to generate machine code for model production on a FDM 3000 RP machine.

Similar to other RP machines, a FDM machine also builds a part layer by layer. The raw material of FDM is in a coil of filament. The filament is feed to a FDM head where the material is melted and extruded through a FDM tip onto the model surface. For each layer, the FDM machine will draw based on a predefined tool path on areas of a cross-section. Upon completion of building one layer, the platform holding the part will move down by one layer and a new layer can be further produced. The process is repeated until the entire part is build. For down facing areas, there is also a need of support that is built with another tool tip on the same FDM head. The support material is a water-soluble material and can be easily melted off in a special liquid solution after model production.

For a model built with adaptive slicing and selective hatching, there are three types of slicing layers as shown in Fig. 5. The first one is a general layer on which all the contour areas are scanned and filled. The second type is a selective hatching layer where only the skin areas are scanned and filled. The last one is the top selective hatching layer where both the skin areas and internal areas will be scanned and filled, however, with different building thicknesses for skin and internal areas, respectively. Fig. 8 shows the building cycle of one selective hatching unit for the realization of adaptive slicing with selective hatching on a FDM machine.

4.1. Adaptive slicing module

The slicing procedure begins with the selection of model surfaces that form an enclosed volume with appropriate surface normal (pointing outward) and slicing parameters. Key slicing parameters include cusp heights, and minimum and maximum layer thickness. The extreme points are first detected with all the input data. The slicing operation will start from the lowest extreme point of the model layer by layer upwards and terminates when reaching the top extreme point.

For each layer, a horizontal \( xy \)-plane at the corresponding \( z \)-height will intersect with all surfaces and the intersection contours are collected and organized as contour points based on a step size. The vertical curvature and the \( z \)-coordinate of the normal vector at each contour point on the corresponding surface are computed for calculating a permissible layer thickness following Appendix A. The layer thickness of the entire slice is determined as the minimum permissible layer thickness of all the contour points or the distance from the current slice to the next peak point, whichever is less. The final layer thickness is truncated by the minimum and the maximum layer thickness of the corresponding RP system. Skin contour coordinates are further calculated based on the final layer thickness and other input parameters following Appendix B.

After the entire slicing procedure, the slicing results and related parameters, including the total number of adaptive layers, positive and negative cusp height, minimum and maximum layer thickness, contour step size and extreme points information, are saved in a file for further processing.

4.2. Selective hatching module

The selective hatching procedure is implemented as a stand-alone SHM. This software module performs four main functions and produces the final SML code for machine model production in cooperation with the Quickslice software of the RP system.
Quantitative hatching. This is the main function for producing the selective hatching results. Given the adaptive slicing results and a wall thickness of the skin region for selective hatching, this function divides all the adaptive slicing layers into selective hatching units and computes the partition contours for each selective hatching unit starting from offset contour generation. The final results are organized layer by layer with layer attributes, layer parameters, external contours, internal contours and the partition contours. The layer attributes contain information such as the type of layer, which could be a general layer, a selective hatching layer, or a top selective hatching layer. Layer parameters include the \( z \)-height and the hatching unit height in case of a top selective hatching layer.

SSL code generation and support integration. This function is mainly used for interfacing with the Quickslice software of the RP system through SSL (Stratasys Slicing Language), a proprietary file format defined by Stratasys for storing contour slicing information [27,28]. This function converts slicing information into a SSL file. The input can be either the adaptive slicing results or the selective hatching contour information from SHM Function #1. For selective hatching contours, this function also integrates the selective hatching results with support structure for model production. The support structure is indirectly produced from the adaptive slicing results using the Quickslice software. A ‘part fast’ option can also be selected in this function for a selective hatching model. This is an option to further reduce building time by applying an air gap and the maximum road width of filament material for internal selective hatching areas.

SSL code conversion for modifying support parameters. Support structures produced using Quickslice software may not directly work with the adaptive slicing data. This function modifies the support parameters of the SSL file and makes it compatible with adaptive slicing. It detects the entire support structure contour group in the SSL file and modifies the support structure into appropriate individual groups.

SML code conversion for adaptive slicing and selective hatching with support definition. SML (Stratasys Modeling Language) is the machine language that is used for communication between the Quickslice software and a FDM machine [27]. This file contains the actual NC codes for controlling the FDM system. The SML code generated by the Quickslice software may not directly work either in case of adaptive slicing with selective hatching due to some incompatible commands for the \( z \)-height. This incompatibility causes the FDM tip one layer higher and therefore the filament material cannot stick to the model. This function resolves this incompatibility problem and produces a correct SML file that can be used by a FDM machine to produce adaptive slicing with or without selective hatching prototype model.

4.3. Model production with the developed software modules

With a NURBS-based CAD model, all surfaces are first adaptively sliced into surface contours using the ASM

Fig. 9. Realization of adaptive slicing with or without selective hatching on a FDM machine.
software module. The selective hatching algorithm is then applied to the adaptive slicing results using the SHM software module. The adaptive slicing and selective hatching information is interfaced with the Quickslice software of Stratasys using the SSL file format. The Quickslice software is finally used to generate the tool path in SML code. The SHM software module is also used to solve a few compatibility problems of SSL and SML files generated by the Quickslice software. Fig. 9 highlights the complete procedure for the realization of adaptive slicing with or without selective hatching on an FDM machine. In this figure, AS stands for adaptive slicing, SH stands for selective hatching and SHM F#1 stands for SHM function #1.

As shown in Fig. 9, the entire procedure starts from adaptive slicing using the ASM. For model production using adaptive slicing only, the adaptive slicing results are further processed by Quickslice for support and SML code generation. The final SML codes are used for producing adaptive slicing models on a FDM RP machine.

For model production using both adaptive slicing and selective hatching, selective hatching areas are further produced from the adaptive slicing information using the SHM. The adaptive slicing and selective hatching results are further integrated with the support structure that is produced using adaptive slicing information only. The Quickslice software is then used for SML code generation. After correction of a few compatibility problems, the SML code can be finally used for producing adaptive slicing and selective hatching models with an FDM RP machine.

5. Case studies and discussions

This section presents several case studies for demonstrating the performance of the developed algorithms. Some analytical results are first presented in Sections 5.1 and 5.2. Experimental results obtained with a FDM RP machine are summarized in Section 5.3.

With the analytical study, two case studies are presented to compare the slicing results of adaptive slicing with and without the use of the selective hatching strategy. Figs. 10 and 13 show the CAD models of the case studies. Each model is sliced with several cusp height values, the wall/skin thickness is fixed at 2.0 mm when using selective hatching strategy, the allowable layer thickness is between $d \in [0.178, 0.508 \text{ mm}]$ and a contour step size of 1.0 mm is used for sampling the contour data. Scanning area reduction of the model that made with/without selective hatching will be compared.

5.1. A mechanical part

The CAD model of the first case study is shown in Fig. 10. The size of the object is about $80 \text{ mm} \times 100 \text{ mm} \times 40 \text{ mm}$. Fig. 11 shows one section of the object parallel to the $xz$-plane and eight slabs are subdivided with the identified peak features. The part is sliced with a pre-specified cusp height tolerance. Fig. 12 shows the slicing results for the case of cusp height $\delta = \pm 0.05 \text{ mm}$ represented by two quantities and with three curves. One of the quantities illustrates the relationship between the number of layers and the $z$ height, a quantity representing

![Fig. 10. The surface model of a mechanical part for case study 1.](image1)

![Fig. 11. Section of the mechanical part in $zx$-plane.](image2)

![Fig. 12. Case study 1e: comparison between slicing results of with and without selective hatching (cusp height $\delta = \pm 0.05 \text{ mm}$).](image3)
the variable layer thickness. Both of the scanning strategies, either with or without selective hatching, share the same curve since the same adaptive slicing algorithm is applied. The other quantity illustrates the relationship between the number of layers and the accumulated total slicing area at that layer, a quantity proportional to the build time required till that layer. Table 1 summarizes the slicing results of this example computed with different cusp height values. Following the results, the number of adaptive slicing layers is greatly reduced compared with that in uniform slicing for all tested tolerances and hence a significant build time saving. However, for this example, no further improvement is obtained with selective hatching. This is mainly because the lower portion of this part was already built with the maximum layer thickness with adaptive slicing and the upper portion of the model is a sell like part with a thin wall thickness. In both cases, no further improvement is possible with selective hatching. It can thus be concluded that for parts without slope surfaces or for parts with thin walls, there is no major advantage in using the selective hatching strategy.

### Table 1
Slicing results of case 1 shown in Fig. 10

<table>
<thead>
<tr>
<th>Quantities</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cusp height (mm)</td>
<td>Total number of adaptive layers</td>
<td>Number of layer reduction (B versus U, %)</td>
<td>Scanning area without selective hatching (×10⁶ mm²)</td>
<td>Scanning area with selective hatching (×10⁶ mm²)</td>
<td>Scanning area reduction (E versus D, %)</td>
<td></td>
</tr>
<tr>
<td>Case 1a</td>
<td>0.15</td>
<td>99</td>
<td>56.0</td>
<td>0.331919</td>
<td>0.331914</td>
<td>0.002</td>
</tr>
<tr>
<td>Case 1b</td>
<td>0.125</td>
<td>103</td>
<td>54.2</td>
<td>0.333800</td>
<td>0.333620</td>
<td>0.05</td>
</tr>
<tr>
<td>Case 1c</td>
<td>0.1</td>
<td>111</td>
<td>50.7</td>
<td>0.337711</td>
<td>0.335138</td>
<td>0.8</td>
</tr>
<tr>
<td>Case 1d</td>
<td>0.075</td>
<td>121</td>
<td>46.2</td>
<td>0.342657</td>
<td>0.338872</td>
<td>1.1</td>
</tr>
<tr>
<td>Case 1e</td>
<td>0.05</td>
<td>131</td>
<td>41.8</td>
<td>0.347952</td>
<td>0.342970</td>
<td>1.5</td>
</tr>
</tbody>
</table>

The number of layers in uniform slicing at \( d = 0.178 \) mm is \( U = 225 \). For all the results in this table, the minimum and the maximum layer thickness are 0.178 and 0.508 mm, respectively. The wall/skin thickness for selective hatching is 2.0 mm and contour step size is 1.0 mm.

The part of the second case study shown in Fig. 13 is a camera model. The size of the part is about 60 mm × 126 mm × 70 mm. Fig. 14 shows the slicing results for cusp height tolerance \( \delta = 0.075 \) mm represented by the same quantities discussed in Section 5.1. In this case, there are a total number of 263 variable layers and a 33.1% layer reduction is achieved compared with uniform slicing at \( d = 0.178 \) mm. The total scanning area is \( 1.127247 \times 10^6 \) and \( 0.892271 \times 10^6 \) mm² for the cases of with and without selective hatching, respectively. The selective hatching strategy provides a further 20.85% reduction in the scanning area compared with the traditional scanning/hatching strategy. Table 2 summarizes the slicing results of this example with different cusp height values. Following the results, significant time reduction can be achieved with adaptive slicing and a further significant time reduction can also be achieved with selective hatching. It can also be concluded that in case of large dense objects or objects with thick walls, significant build time reduction can be achieved.

5.2. A camera model

![Fig. 13. A camera surface model for case study 2.](image)

![Fig. 14. Case study 2d: comparison between the slicing results of with and without selective hatching (\( \delta = 0.075 \) m).](image)
be achieved with selective hatching. It is also obvious from Table 2 that the use of adaptive slicing is most effective for low tolerance requirement, while the use of selective hatching is most effective for tight tolerance requirement, as long as the minimum layer thickness is required for certain area. A combined use of adaptive slicing and selective hatching would thus offer consistently effective build time reduction independent of tolerance requirement. One could therefore afford to use a tight tolerance and hence improved surface quality.

<table>
<thead>
<tr>
<th>Quantities</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cusp height (mm)</td>
<td>Total number of adaptive layers</td>
<td>Number of layer reduction (B versus U, %)</td>
<td>Scanning area without selective hatching ($\times 10^6$ mm$^2$)</td>
<td>Scanning area with selective hatching ($\times 10^6$ mm$^2$)</td>
<td>Scanning area reduction (E versus D, %)</td>
</tr>
<tr>
<td>Case 2a</td>
<td>±0.15</td>
<td>169</td>
<td>57.0</td>
<td>0.732612</td>
<td>0.732544</td>
<td>0.01</td>
</tr>
<tr>
<td>Case 2b</td>
<td>±0.125</td>
<td>185</td>
<td>52.9</td>
<td>0.795916</td>
<td>0.780504</td>
<td>1.9</td>
</tr>
<tr>
<td>Case 2c</td>
<td>±0.1</td>
<td>217</td>
<td>44.8</td>
<td>0.924723</td>
<td>0.812046</td>
<td>12.2</td>
</tr>
<tr>
<td>Case 2d</td>
<td>±0.075</td>
<td>263</td>
<td>33.1</td>
<td>1.127247</td>
<td>0.892271</td>
<td>20.9</td>
</tr>
<tr>
<td>Case 2e</td>
<td>±0.05</td>
<td>315</td>
<td>19.8</td>
<td>1.383967</td>
<td>0.922295</td>
<td>33.4</td>
</tr>
</tbody>
</table>

The number of layers in uniform slicing at $d = 0.178$ mm is $U = 393$. For all the results in this table, the minimum and the maximum layer thickness are 0.178 and 0.508 mm, respectively. The wall/skin thickness for selective hatching is 2.0 mm and contour step size is 1.0 mm.

Fig. 15. Implementation and realization for efficient prototyping of the camera model with adaptive slicing only: (a) the adaptive skin contours; (b) a close-up front view of the adaptive layers; (c) the produced physical model and (d) a close-up view of the physical model. (See Table 3 for building parameters and building time.)
5.3. Model production on a FDM rapid prototyping machine

The case studies discussed in the previous subsections have been successfully built on a FDM RP machine using adaptive slicing and adaptive slicing with selective hatching. Fig. 15 provides some illustrations for efficient prototyping of the camera model with adaptive slicing. Fig. 15(a)–(c) shows the computed adaptive skin contours and Fig. 15(d) and (e) shows the produced physical prototype. In this case, the cusp height used is \(0.075\) mm and the adaptive layers vary between \(0.178\) and \(0.508\) mm. The total building time is \(28.25\) h (excluding \(2.3\) h support building time). Compared with prototyping with uniform layers of \(0.178\) mm, which takes \(71.12\) h, a total actual building time reduction of \(60.28\%\) is achieved.

Fig. 16(a) and (b) shows the produced prototype models of the mechanical part and the camera model built with adaptive slicing with selective hatching. Fig. 16(c) and (d) shows the prototype models of a torus part built with adaptive slicing and adaptive slicing with selective hatching, respectively. Fig. 16(c) and (d) also shows the support of the torus part that can be easily removed by dipping the part with support in a special solution. The size of the torus model is about \(60\) mm \(\times\) \(130\) mm \(\times\) \(130\) mm and it was also used in Ref. [16] as a test model. In case of the camera model again, the build time for producing the physical model with adaptive slicing and selective hatching is \(17.7\) h. Compared with prototyping with adaptive slicing only, a further build time reduction of \(38.18\%\) is achieved. The overall building time reduction in case of the camera model using adaptive slicing with selective hatching is \(75.11\%\) compared with uniform slicing, a significant improvement achieved. Table 3 summarizes the parameters used for producing the physical prototype models and a comparison among three different approaches for all these three prototype models.

![Fig. 16. Various prototype models produced with adaptive slicing and selective hatching for (a), (b) and (d) and adaptive slicing only for (c). (See Table 3 for building parameters and building time).](image-url)
### Table 3

Comparison between model prototyping with uniform slicing, adaptive slicing and adaptive slicing with selective hatching

<table>
<thead>
<tr>
<th>General parameters</th>
<th>Mechanical part</th>
<th>Camera Model</th>
<th>Torus part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model size (mm × mm × mm)</td>
<td>80 × 100 × 40</td>
<td>60 × 126 × 70</td>
<td>60 × 130 × 130</td>
</tr>
<tr>
<td>Cusp height (mm)</td>
<td>± 0.05</td>
<td>± 0.075</td>
<td>± 0.1</td>
</tr>
<tr>
<td>Number of layers (uniform layer thickness/adaptive layer thickness)</td>
<td>225/131</td>
<td>393/263</td>
<td>730/447</td>
</tr>
</tbody>
</table>

**Building time**

<table>
<thead>
<tr>
<th></th>
<th>Mechanical part</th>
<th>Camera Model</th>
<th>Torus part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform slicing (0.178 mm)</td>
<td>36.87 h</td>
<td>71.12 h</td>
<td>123.82 h</td>
</tr>
<tr>
<td>Adaptive slicing (0.178–0.508 mm)</td>
<td>6.53 h</td>
<td>28.25 h</td>
<td>37.50 h</td>
</tr>
<tr>
<td>Adaptive slicing plus selective hatching</td>
<td>6.47 h</td>
<td>17.70 h</td>
<td>24.73 h</td>
</tr>
</tbody>
</table>

**Building time reduction**

<table>
<thead>
<tr>
<th></th>
<th>Mechanical part</th>
<th>Camera Model</th>
<th>Torus part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive slicing over uniform slicing (%)</td>
<td>− 82.29</td>
<td>− 60.28</td>
<td>− 69.71</td>
</tr>
<tr>
<td>Adaptive slicing plus selective hatching over adaptive slicing (%)</td>
<td>− 0.92</td>
<td>− 37.35</td>
<td>− 34.05</td>
</tr>
<tr>
<td>Adaptive slicing plus selective hatching over uniform slicing (%)</td>
<td>− 82.45</td>
<td>− 75.11</td>
<td>− 80.03</td>
</tr>
</tbody>
</table>

*a* Excluding support building time.

Following the results summarized in Table 3, it can be concluded that adaptive slicing is an effective approach in reducing the build time for any parts having freeform features or vertically sloped surfaces. With the case studies presented in this paper, a build time reduction of 82.29, 60.28 and 69.71% are achieved, respectively, compared with uniform slicing. With selective hatching, a further improvement can be made for freeform parts having thick walls. With the presented case studies, a further build time reduction of 37.35 and 34.05% are achieved for the camera model and the torus model, respectively, with selective hatching. In case of the mechanical part, a build time reduction of 82.29% has been achieved with adaptive slicing along and no further reduction is achieved with selective hatching. As discussed in Section 5.1, this is mainly because with adaptive slicing the main body of the part, i.e. the lower part of the model, was already built with the maximum layer thickness. In addition, selective hatching may not be effective in case of parts with just thin walls. For the upper portion of the mechanical part, it is just a spherical shell with a wall thickness and, hence, may not be effective to further apply selective hatching. However, the overall build time reduction with adaptive slicing and selective hatching combined is consistently significant irrespective of individual parts, as long as there are small or sloped features that require a small layer thickness. The overall build time reduction amounts to 82.45, 75.11 and 80.03%, respectively, for the present case studies.

When comparing the actual building time reduction with the scanning area reduction of the models discussed in this section for selective hatching, one may find that for the same slicing parameters, the percentage of actual time reduction in the experiments is better than that of simulation presented in Section 5.2 for the camera model. The computed scanning area of Tables 1 and 2 is proportional to the build time required, but not exactly a quantity for measuring the actual build time. The actual build time also depends on a particular process and scanning strategy used. For FDM prototyping, the roadwidth, i.e. the width of the wire when pushed on top of the model surface, is usually two times the layer thickness. In other words, for thick layers, we not only saved time in building less number of layers, but also saved time when scanning the same surface area due to the use of a larger road width for thick layers. In addition, a part fast option was also used in the experiments for interior areas, which can further reduce the scanning time by allowing a small air gap between neighboring wires/roads.

### 6. Conclusions

This paper incorporates a selective hatching strategy with an adaptively slicing algorithm for tackling the compromising issue of building speed and surface accuracy in RP. New adaptive slicing and selective hatching algorithms with arbitrary tolerance control have been developed for the determination of both adaptive layer thickness and skin contour points. The adaptive slicing algorithm has been implemented in a Unigraphics software environment and the selective hatching algorithm was implemented as a stand-alone software module that works collaboratively with the Quickslice software of the RP system. Several case studies are presented to evaluate the efficiency of the developed adaptive slicing and selective hatching algorithms. The models have been successfully built on a FDM RP machine with both adaptive slicing and adaptive slicing with selective hatching. Both of the approaches are effective in reducing the overall build time, while maintaining the desired surface quality. The results show that while adaptive slicing is effective for any complex model with freeform or slopped surfaces, the selective hatching strategy is most effective for general freeform or slopped parts with thick walls. A combined use
of adaptive slicing and selective hatching also leads to consistently significant build time reduction irrespective of the tolerance required and individual parts concerned, as long as the part has small features or sloped surfaces that require a small layer thickness, and hence improved surface quality. While the algorithms presented in the paper were only realized for FDM RP, the developed algorithms are applicable to many other RP processes, such as stereolithography, SLS, laminated object modeling with multiple layers and similar processes. Further work will be conducted to investigate the physical properties of the parts built with the developed approaches. The developed algorithms for adaptive slicing with arbitrary tolerance control and the general algorithm for selective hatching can also be extended for processing models represented in STL format.

Acknowledgements

The work described in this paper is supported by a Strategic Research Grant (Project No. 7001296) from City University of Hong Kong.

Appendix A. Layer thickness determination

Given a contour point \( \mathbf{P} = [x, y, z]^T \) and let \( \delta_n, \delta_p \) and \( \delta = \delta_n + \delta_p \) be the negative, positive and total allowed cusp height, respectively, \( d \) be the next layer thickness, \( \mathbf{n} = [n_x, n_y, n_z]^T \) be the unit vector of surface normal at \( \mathbf{P} \), and, \( \rho \) be the radius of the normal curvature \( \kappa_n \) at a point \( \mathbf{P} \) along the vertical direction with \( \rho = 1/\kappa_n \) (see Appendix C for the evaluation of \( \kappa_n \)). The determination of layer thickness with either negative tolerance or positive tolerance can be found in Ref. [11]. For mixed tolerances, Fig. 4 can be used to illustrate various cases and derive the corresponding formulas for layer thickness determination. The results are as follows:

\[
d = +\frac{\delta}{n_z} \quad \text{(A1)}
\]

\[
d = -\frac{\delta}{n_z} \quad \text{(A2)}
\]

\[
d = +\rho n_z + \sqrt{\rho^2 n_z^2 + \delta(2\rho + \delta_p - \delta_n)} \quad \text{(A3)}
\]

\[
d = -\rho n_z + \sqrt{\rho^2 n_z^2 + \delta(2\rho - \delta_p + \delta_n)} \quad \text{(A4)}
\]

\[
d = +\rho n_z - \sqrt{\rho^2 n_z^2 - \delta(2\rho + \delta_p - \delta_n)}; \quad \text{if } \rho^2 n_z^2 > \delta(2\rho + \delta_p - \delta_n) \quad \text{(A5)}
\]

\[
d = -\rho n_z - \sqrt{\rho^2 n_z^2 - \delta(2\rho - \delta_p + \delta_n)}; \quad \text{if } \rho^2 n_z^2 > \delta(2\rho - \delta_p + \delta_n) \quad \text{(A6)}
\]

\[
d = +2\rho n_z; \quad \text{if } \rho^2 n_z^2 \leq \delta(2\rho + \delta_p - \delta_n) \quad \text{(A7)}
\]

The two special cases of Eqs. (A7) and (A8) when \( n_z \) is very small or the point \( \mathbf{P} \) lies very close to the equator are shown by Fig. 4(g) and (h), respectively. Eqs. (A1)–(A8) are the formulas for layer thickness determination at a point. To find the layer thickness for the entire slice, the minimum layer thickness at all contour points on that layer is used. If there are more than one surfaces on the current slice, all the resulting layer thickness must be compared and the smallest one should be used. We check all the discrete sample contour points on that layer to determine the minimum layer thickness.

Appendix B. Skin contour evaluation

When the layer thickness \( d \) is obtained for each layer, we can further determine the skin contour points according to the actual cusp height. Let \( \mathbf{P} = [x, y, z + d]^T \) be the skin contour point corresponding to contour point \( \mathbf{P} = [x, y, z]^T \) and let \( \delta_n, \delta_p \), and \( \delta = \delta_n + \delta_p \) be the actual negative, positive and total cusp height, respectively, with \( \delta_n = (\delta_n/(\delta_n + \delta_p))\delta \) and \( \delta_p = (\delta_p/(\delta_n + \delta_p))\delta \). Let us further denote \( n'_x = n_x/\sqrt{n_x^2 + n^2} \) and \( n'_y = n_y/\sqrt{n_y^2 + n^2} \). The formulas for determining the skin point at \( \mathbf{P} \) can also be derived from Fig. 4 and can be represented in terms of either \( \delta_n \) or \( \delta_p \). The following equations define skin contour points in terms of \( \delta_n' \):

\[
\begin{align*}
x(x) &= x - \frac{n'_x}{\sqrt{1 - n^2}} \delta_n' \\
y(y) &= y - \frac{n'_x}{\sqrt{1 - n^2}} (dn_z + \delta_n') \\
x(x) &= x - \frac{n'_x}{\sqrt{1 - n^2}} (dn_z + \delta_n') \\
y(y) &= y - \frac{n'_x}{\sqrt{1 - n^2}} (dn_z + \delta_n')
\end{align*}
\]

\[
\begin{align*}
x(x) &= x - \rho n_z + n'_x\sqrt{(p - \delta_n')^2 - \rho^2 n_z^2} \\
y(y) &= y - \rho n_z + n'_x\sqrt{(p - \delta_n')^2 - \rho^2 n_z^2}
\end{align*}
\]

\[
\begin{align*}
x(x) &= x + \rho n_z - n'_x\sqrt{(p + \delta_n')^2 - (p - d)^2} \\
y(y) &= y + \rho n_z - n'_x\sqrt{(p + \delta_n')^2 - (p - d)^2}
\end{align*}
\]

\[
\begin{align*}
x(x) &= x - \rho n_z + n'_x\sqrt{(p - \delta_n')^2 - (p - d)^2} \\
y(y) &= y - \rho n_z + n'_x\sqrt{(p - \delta_n')^2 - (p - d)^2}
\end{align*}
\]
\[
\begin{align*}
\begin{aligned}
x_u &= x + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
y_u &= y + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
x_v &= x - m_v + n_v (p - \delta_v) \\
y_v &= y - m_v + n_v (p - \delta_v)
\end{aligned}
\end{align*}
\]  

(B14)

\[
\begin{align*}
x_u &= x + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
y_u &= y + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
x_v &= x - m_v + n_v (p - \delta_v) \\
y_v &= y - m_v + n_v (p - \delta_v)
\end{aligned}
\]  

(B15)

\[
\begin{align*}
x_u &= x + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
y_u &= y + m_u - n_u \sqrt{(p + \delta_u)^2 - p^2 n_z^2} \\
x_v &= x - m_v + n_v (p - \delta_v) \\
y_v &= y - m_v + n_v (p - \delta_v)
\end{aligned}
\]  

(B16)

Appendix C. Vertical curvature evaluation for NURBS surfaces

A NURBS surface [25] is defined by the following equation

\[
S(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}
\]

(C17)

\[
0 \leq u, v \leq 1. \text{ In the above equation, } p \text{ and } q \text{ are the degrees of the surface along } u \text{ and } v \text{ directions, respectively, } P_{i,j} \text{ for } i = 1, 2, \ldots, m + 1 \text{ and } j = 1, 2, \ldots, n + 1 \text{ are the } (m + 1) \times (n + 1) \text{ control points, and } N_{i,p}(u) \text{ for } i = 0, 1, \ldots, m \text{ and } N_{j,q}(v) \text{ for } j = 0, 1, \ldots, n \text{ are normalized } B \text{-splines that are uniquely defined by a set of } u \text{-knots and } v \text{-knots, respectively. The normal curvature of a general surface } S(u, v) \text{ can be found in Ref. [11] as follows}
\]

\[
\kappa_n = \frac{L + 2M_{\text{ortho}} + N \gamma^2}{E + 2F_{\text{ortho}} + G \gamma^2}
\]

(C18)

where \(L = \sum_{m} \sum_{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} \), \(E = \sum_{m} \sum_{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} \), \(F = \sum_{m} \sum_{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} \), \(G = \sum_{m} \sum_{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} \), and \(\gamma = \frac{\partial^2 S(u, v)}{\partial u \partial v} \). The quantity \(\gamma = \frac{\partial^2 S(u, v)}{\partial u \partial v} \) represents a particular direction in the tangent plane of the current contour point and \(\gamma_{\text{ortho}} \) is another vector on the same tangent plane orthogonal to \(\gamma \). If we select \(\gamma \) being the tangent direction of the current contour at the point for evaluation, \(\gamma_{\text{ortho}} \) then defines the vertical direction for normal curvature evaluation. The evaluation of derivatives of general NURBS curves and surfaces can be found in Ref. [25]. Based on the derivatives of NURBS geometry, one obtains the following equation for \(\gamma \)

\[
\gamma = \left( \frac{\partial S}{\partial u} \right)_{\text{at level } z_0} = \left( \frac{\partial S}{\partial u} \right)_{\text{at level } z_0} = \frac{Z_u(u, v) - z_0 w_u(u, v)}{Z(u, v) - z_0 w_u(u, v)}
\]

(C20)

where \(Z(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} z_i \) and \(w(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(u) N_{j,q}(v) w_{i,j} \) stand for the \(z \)- and \(w \)-coordinates of the NURBS surface in homogeneous space and \(z_0 \) is the \(z \)-coordinate of the current slice in three-dimensional space.

References


Weiyn Ma is an Associate Professor in manufacturing engineering at City University of Hong Kong (CityU). He lectures in the areas of geometric modeling, CAD/CAM and rapid prototyping. His present research interests include computer aided geometric design, CAD/CAM, virtual design and manufacturing, rapid prototyping and reverse engineering. Prior to joining CityU, he worked at Materialise NV, a rapid prototyping firm in Belgium. He also worked in the Department of Mechanical Engineering of Katholieke Universiteit Leuven (K.U. Leuven), Belgium, and the Department of Space Vehicle Engineering of East China Institute of Technology (ECIT), Nanjing, China. He obtained a BSc in 1982 and an MSc in 1985 from ECIT, an MEng in 1989 and a Ph.D. in 1994 from K.U. Leuven.

Wing-Chung But is a research assistant and research student in the Department of Manufacturing Engineering at City University of Hong Kong. He obtained a BEng degree (Bachelor of Engineering in Manufacturing Engineering) and a M.Phil degree (Master of Philosophy) at the same University in 1999 and 2002, respectively. His research interests include CAD/CAM, product design, software engineering and rapid prototyping.

Peiren He is currently with Nanoscale Technology and Engineering Laboratory (NTEL) of the Department of Mechanical Engineering at the University of Alberta, Canada. His current research interests include CAD/CAM in micro/nano-structures. His principal former affiliation was with the Department of Manufacturing Engineering at Nanjing University of Science and Technology (NUST), China. He also worked in the Department of Manufacturing Engineering and Engineering Management at City University of Hong Kong as a research staff. He received a B.Sc. in 1985 and an M.Sc. in 1993 from NUST, a Ph.D. in 2003 from University of Saskatchewan, Canada.