






Efficient and Valid Surface Reconstruction for Workpiece Models in Frame-Sliced Voxel Based Machining Simulation

Jimin Joy¹ , Jack Szu-Shen Chen²  and Hsi-Yung Feng³ 

¹The University of British Columbia, jjoy@alumni.ubc.ca

²The University of British Columbia, jsschen38@gmail.com

³The University of British Columbia, feng@mech.ubc.ca

Corresponding author: Hsi-Yung Feng, feng@mech.ubc.ca

Abstract. This paper presents an efficient method to reconstruct 2-manifold triangle mesh surfaces for machined workpiece models in the frame-sliced voxel representation (FSV-rep) format. The FSV-rep workpiece models have been employed to attain superior computational efficiency in machining simulation with the simulated workpiece model geometry output as a triangle mesh to ensure accuracy. Generating the triangle mesh from the simulated workpiece model was done using algorithmic methods that did not offer the best computational efficiency. In this paper, practical simplifications have been made to enable the use of a finite number of pre-defined sliced voxel shapes for the FSV-rep workpiece model. In particular, a comprehensive 22-case lookup table has been derived to cover all the possible basic frame-sliced voxel shapes without using any non-geometric information, unlike the existing lookup table-based methods. The applicability of the derived lookup table to generate closed 2-manifold triangle meshes for the FSV-rep workpiece models has been validated via a series of machining test cases. Quantitative comparisons with a typical algorithmic method with regards to computational time have been made to illustrate the computational advantages of the presented lookup table-based method.

Keywords: Machining simulation; workpiece surface reconstruction; lookup table

DOI: <https://doi.org/10.14733/cadaps.2021.1327-1340>

1 INTRODUCTION

Geometric simulation is an essential technical component in the emerging virtual and intelligent machining technologies [1],[9]. Visualization of the simulated workpiece geometry is one of the main functions in the human-computer interface of the machining simulation system. It helps to inspect the simulated workpiece geometry and more importantly, to verify the associated machining program to be used by a machine tool to produce the physical part [4],[11]. Since the purpose of simulation is to generate the machined workpiece geometry whereas that of visualization is to display the simulated workpiece, the underlying geometric modeling format

suitable for simulation is not necessarily the same as the format suitable for visualization. In fact, the model suitable for simulation is expected to be capable of being dynamically changed and constantly updated. On the other hand, the model suitable for visualization can be static in terms of the model data but is expected to display the machined workpiece with satisfactory visual realism.

A new and effective geometric modeling format, named as the frame-sliced voxel representation (FSV-rep), has recently been introduced to model the changing workpiece geometry for general milling operations [5]. It has been demonstrated that FSV-rep offers both modeling accuracy and memory efficiency as well as computational efficiency [6]. In addition to model accuracy, memory efficiency and computational efficiency of FSV-rep in machining simulation, fast visualization of the FSV-rep workpiece model is an equally important function in practical applications. Since modern graphics processing technologies favor the triangle mesh surface representation, the main task for the fast visualization of an FSV-rep workpiece model is then to quickly generate a triangle mesh representation for it. The method to be presented in this paper utilizes a comprehensive lookup table for the fast generation of the triangle mesh representation.

In the subsequent sections of this paper, the existing surface generation methods employed in machining simulation to visualize the simulated workpiece model will be reviewed first. The salient features of an FSV-rep model that should be considered in the development of an effective surface generation method will then be presented along with the derived lookup table for fast triangle mesh generation. The applicability of the lookup table in all machining scenarios will be established and the implementation details given. The superior computational performance of the lookup table-based method compared to an existing algorithmic method will be demonstrated by a set of test cases involving a wide variety of workpiece shapes.

2 RELEVANT EXISTING METHODS

Effective representation of geometric information in a virtual machining simulation environment is essential to the visualization of the simulated workpiece model geometry [12]. Visualization by surface rendering of workpiece models represented as NURBS or polyhedral B-rep solid models is a trivial task since the definitions of these solid models are already based on boundary surface elements (although NURBS B-rep solid models are often tessellated into triangle mesh representations to facilitate visualization). Surface rendering of non-B-rep workpiece models for visualization is a more involved task. For example, vector and space partitioning workpiece models both need to generate applicable boundary surface representations in order to visualize the models. For voxel-based space partitioning workpiece models, dedicated volume rendering hardware was proposed as an applicable option to visualize the voxel models [13]. However, such specialized hardware has not gained popularity due to the added cost. Hence, an effective surface generation technique for voxel models is still the preferred option.

Spatial range data collected from 3D scanning or medical imaging have been processed into a voxel model representation first and then converted to a surface model for visualization using the widely used marching cubes algorithm [10]. The associated model conversion relies upon the specific distance field value at each corner point of a voxel to create the surface patches within the voxel via an interpolation technique. For binary voxel models with information available only to infer the corner occupancy status, the surface generation method needs to approximate the surface patch using mid-points of the voxel edges as patch boundary vertices [3]. This can add undesirable deviations onto the generated surface mesh. Point cloud triangulation from the center points of the surface voxels is another technique that has been used.

The marching cubes algorithm is computationally fast once the corner occupancy status of the voxel has been identified. This is because it utilizes a lookup table of pre-defined triangulations for all possible configurations of voxel corner point occupancy. The number of such configurations is finite and has many symmetry groups. The resulting lookup table contains 15 unique

configurations [10]. However, for some configurations, there can be conflicting triangulations at the shared face of neighboring voxels, causing the resulting triangle mesh to have holes and become non-manifold. To handle these situations, improved lookup tables and robust implementations have been developed [2],[8]. These improved techniques need additional non-geometric information such as spatial range data at the voxel corner points and are, therefore, not directly applicable to binary voxel models from simulation. Any technique that is to be devised to correct the issue will involve further computations. This will affect the overall computational efficiency of the mesh generation process from the FSV-rep model which only contains geometric information.

Surface construction methods for vector-based workpiece models for machining simulation are also worth considering. Many of these methods are only applicable to the Z-map or dixel models [7],[16],[17]. Ren et al. [14] developed an algorithmic approach to triangulate surface patches within grid cells of a tri-dixel workpiece model. The algorithmic approach is robust and can handle all practical cases. It represents a good alternative to the marching cubes approach to generate boundary surfaces for voxel models as well. However, because a triangle mesh patch has to be generated for each involved voxel, a computationally fast method is preferred, in particular for machining animation where the triangulation computation has to be done very fast in order to provide the needed display frames per second. The lowest display requirement of 15 frames per second leaves only about 66 milliseconds to generate a frame.

3 BOUNDARY SURFACE GENERATION FOR FSV-REP MODELS

FSV-rep is a recently developed workpiece model representation format to facilitate machining simulation. It is formulated to enable efficient and accurate update of the simulated workpiece model. The newly introduced frame-sliced voxels enable sub-voxel modeling accuracy by using frame-crossing points to refine the voxels crossing the surface of the model, referred to as the surface voxels. As shown in Figure 1, frame-crossing points designate the locations where the workpiece surface crosses the edge-frame of each surface voxel of the workpiece model. To achieve memory efficiency for the voxel model, the voxels of the workpiece model are defined using two levels: the coarse and fine voxels. The coarse voxels are used to model the bulk of workpiece volume with a very light memory requirement. Fine voxels are used only within each coarse voxel on the model surface to provide smaller surface voxels, thereby improving the effective voxel model resolution. With the frame-crossing points, the fine surface voxels are transformed into frame-sliced voxels as depicted in Figure 1(c), which provide the ability to represent the workpiece geometry with sub-voxel accuracy.

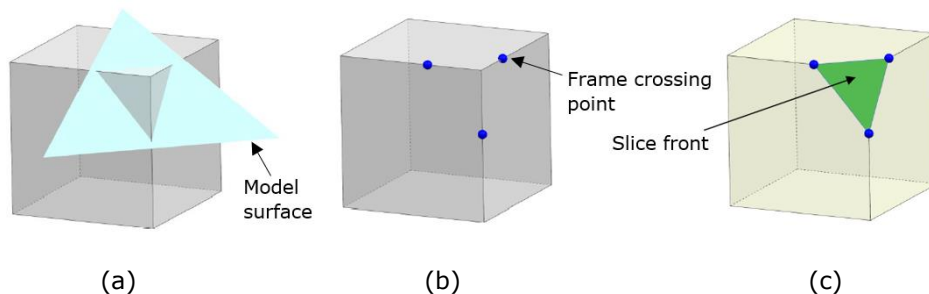


Figure 1: Frame-sliced voxel: (a) surface voxel crossing a model surface; (b) frame-crossing points at the intersection of the surface voxel and model surface; and (c) slice-front from the frame-crossing points, resulting in the frame-sliced voxel.

An FSV-rep model can be viewed as a collection of frame-sliced fine surface voxels for the model surface mesh reconstruction purpose. Each sliced fine surface voxel contains a set of sliced faces

referred to as the slice-front. A slice-front in effect represents the visible small local model surface patch associated with the corresponding fine surface voxel. All such slice-fronts together form a 2-manifold surface representation for the FSV-rep model.

The geometry of the simulated workpiece model can be fully evaluated and inspected for process verification only when a quality visual representation is available. For effective model visualization, the 2-manifold surface from the sliced-fronts of the FSV-rep model is to be transformed into a triangle mesh representation. The triangle mesh representation is chosen as it is a highly versatile geometric representation format. Although there are generally a large number of constituent triangle elements in a triangle mesh, the related computationally intensive rendering task can be efficiently performed in parallel by modern graphics processing units to yield acceptable computational time.

Apart from model visualization, a triangle mesh representation for an FSV-rep workpiece model has many other merits. First, triangle mesh is a well-established and matured geometric model representation format for which many data analysis and processing algorithms are readily available. Machining error determination for the simulated workpiece model is one such task which can be easily performed by comparing with a reference triangle mesh model. Model data transfer to other computer-aided design and modeling tools for making design changes is another common scenario. It should be noted that a closed 2-manifold triangle mesh representation of the simulated workpiece model is targeted in this work so that the subsequent data analysis and processing tasks such as machining error identification and machined sharp edge restoration [15] can be completed successfully.

In this section, a lookup table-based method to rapidly generate a 2-manifold triangle mesh from an FSV-rep workpiece model is to be presented. The method was inspired by the classic marching cubes technique [10] and extended to suit workpiece models generated in machining simulation with pure geometric information alone. Some basic assumptions about the workpiece model geometry are to be made first to help resolve ambiguous cases that can arise while using the lookup table-based method. It will be seen later in the case studies section that these assumptions are valid and do not impact the simulated workpiece geometry.

3.1 Basic Assumptions

The first assumption is that there will be no thin through gaps for any sliced fine surface voxel on the FSV-rep workpiece model due to the movement of a milling tool. These through gaps are depicted in Figure 2 and have the gap width smaller than the edge length of the fine voxel. In other words, no fine voxel will need to be split into two parts. Since the fine voxel edge length should always be set to be much smaller than the milling tool diameter, the assumption of no through gaps within the sliced fine surface voxels will not lead to any global geometry deviation for the FSV-rep workpiece model.

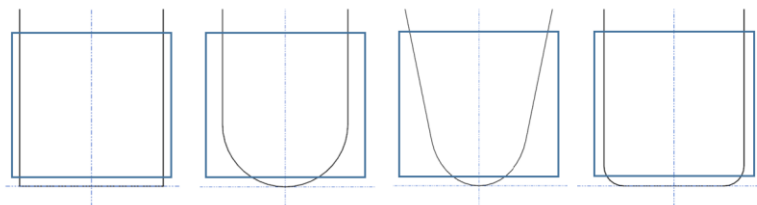


Figure 2: Through gaps created by different types of milling cutters inside voxels of a comparatively large size.

The second assumption relates to the thin machined features on the workpiece. These thin features are assumed to be wide enough to include more than one fine voxel along the feature

width direction. This assumption is made to ensure that thin machined features can be properly captured and constructed by the presented lookup table-based method. Such a requirement can be met by setting a small fine voxel grid spacing relative to the expected width of the thin machined feature.

3.2 Frame-sliced Voxel Configurations

The FSV-rep workpiece model has some favorable features that facilitate the generation of a closed 2-manifold triangle mesh representation. First, with the composition of independent frame-sliced fine surface voxels making up the boundary of the complete model, the triangulation can be performed independently for each frame-sliced voxel. Second, each frame-sliced voxel stores the frame details as frame-crossing points present on the frame edges as shown in Figure 1. For clarity, three related terms are defined here with some depiction in Figure 3: a *frame edge* is the fixed line segment between two voxel corner points; *frame edge segments* are active portions of a frame edge after a cutting tool operates on a frame edge and remove a portion of it; and a *frame edge configuration* refers to the composition of all the frame edge segments lying on the corresponding frame edge.

For a frame-sliced fine surface voxel, each of its 12 frame edges can have up to two frame-crossing points as shown in Figure 3. Since the frame-crossing points are stored as a pair for each frame edge, they can be used to infer the orientation of the surface patch within the voxel [5] which will be detailed in Section 5. For a voxel which is within the volume of a workpiece model, all the eight corner points are present but none of the frame crossing points are there. When a tool cuts out portion of the frame edge, the frame crossing points are added for the frame edge. Considering the status (being present or absent) of the frame-crossing points as well as the status of the two end points of a frame edge, there are 16 theoretically possible configurations for a frame edge. Among the 16 theoretically possible configurations, there are only 6 practically viable ones as depicted in Figure 3. As a result, there are 6^{12} theoretically possible configurations for all the 12 frame edges forming a frame-sliced voxel.

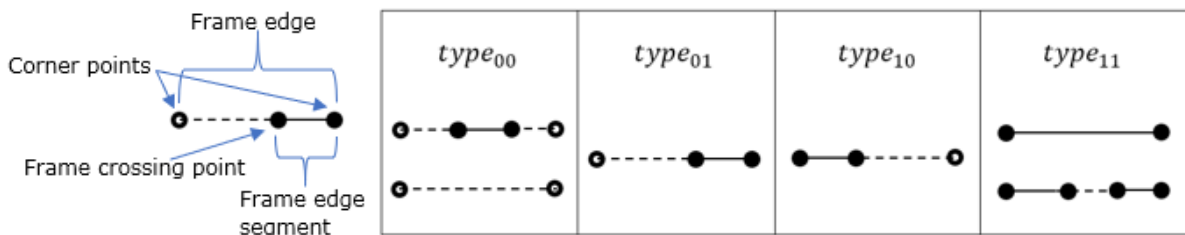


Figure 3: Six practically viable frame edge configurations with up to two frame-crossing points grouped by end point status types.

Out of the massive 6^{12} theoretical possibilities, the frame-sliced voxel configurations that can occur are limited by the following condition:

Condition 1: For any corner point on a particular frame-sliced voxel frame, either three or none of the frame edge segments will be connected to the corner point.

This condition can be easily recognized by noting that it will need all three frame edge segments incident at a corner point to retain a volume for the corner point. Absence of the corner point means that there is no associated volume and thus, no frame edge segments should connect to the corner point.

Condition 1 gives an alternate way to characterize the frame-sliced voxel from the perspective of corner points being active or inactive. Irrespective of the configuration of a frame edge, each of

the 8 corner points can only be active with three connected edge segments (from the frame-sliced voxel in consideration) or inactive with no connected edge segment. Hence, there are a total of $2^8 = 256$ configurations with respect to the corner point status. However, each of the 256 configurations can have many possible combinations of the 12 frame edge configurations under the limitation of Condition 1. For instance, for the configuration of all corner points being inactive, two of the many possibilities for the associated frame-sliced voxel are shown in Figure 4.

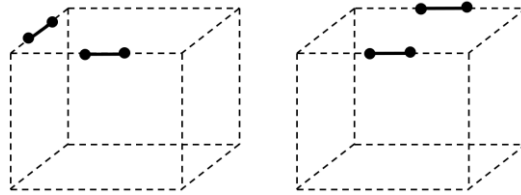


Figure 4: Two sample configurations for a frame-sliced voxel with all corners inactive.

Same as the configuration above, all of the other 255 corner point status configurations also have multiple permitted voxel configurations. This is because for a given frame edge with the status of its two end points specified, there can be more than one possible frame edge configuration as depicted in Figure 3. However, the set of 6^{12} theoretically possible frame-sliced voxel configurations do not consider the limitation of Condition 1. To derive the number of practically possible frame-sliced voxel configurations, the configuration of each frame edge is to be designated as one of the four end point status types: $type_{e_{00}}$, $type_{e_{01}}$, $type_{e_{10}}$ and $type_{e_{11}}$. Here, $type_{e_{00}}$ has both end points inactive; $type_{e_{01}}$ has the left end point inactive and the right end point active; $type_{e_{10}}$ has the left end point active and the right end point inactive; and $type_{e_{11}}$ has both end points active. The 6 frame edge configurations depicted in Figure 3 have been grouped according to the four end point status types. It can be seen that $type_{e_{00}}$ and $type_{e_{11}}$ frame edge configurations have two possibilities each and $type_{e_{01}}$ and $type_{e_{10}}$ have one possibility each.

For a given corner point status configuration for a frame-sliced voxel, the number of frame edge configurations falling into each end point status type is readily known and can be tallied as n_{00} , n_{01} , n_{10} and n_{11} with $n_{00} + n_{01} + n_{10} + n_{11} = 12$. Hence, the number of possible frame-sliced voxel configurations $n_{configs}$ for the given corner point status configuration can be expressed as:

$$n_{configs} = 2^{n_{00}} \times 1^{n_{01}} \times 1^{n_{10}} \times 2^{n_{11}} \quad (3.1)$$

With equation (3.1), the total number of frame-sliced voxel configurations for all the 256 corner point status configurations $N_{configs}$ is then:

$$N_{configs} = \sum_{i=1}^{256} n_{configs}^i \quad (3.2)$$

where $n_{configs}^i$ is the number of frame-sliced voxel configurations corresponding to the i^{th} corner point status configuration. $N_{configs}$ has been evaluated and found to be 36,450. This number is significantly smaller than 6^{12} but still a very large number.

To drastically reduce $N_{configs}$ in order for the size of the associated lookup table to be small enough for practical implementation, the floating segment and isolated gap on a frame edge are to be ignored. As the fine voxel size is quite small, the floating segment and isolated gap only contribute to minor machined surface details such as slight dents and pointed features and are not part of the primary machined faces. Hence, ignoring such segments/gaps will not affect the global surface geometry of the modeled workpiece. The simplified frame edge configurations, in essence, result in the situation where there is only one frame-sliced voxel configuration corresponding to each corner point status configuration. This is because the possible variants within $type_{e_{00}}$ as well

as $type_{11}$ have been reduced to 1 and in equation (1), $n_{configs} = 1$ for all the 256 corner point status configurations.

3.3 Derived Lookup Table

Existing boundary surface generation methods for a given volume have followed either a lookup table-based approach or an algorithmic approach to generate a triangle mesh from a voxel-based model. The reported lookup tables [2],[8],[10] are very efficient as they match each specific voxel configuration with a corresponding triangulation result or voxel shape. However, ambiguity may occur for the original 15-case lookup table and the improved lookup tables require extra non-geometric information not readily available in current grid-based geometric machining simulation workpiece modeling formats such as the FSV-rep model. As for the algorithmic approach, the involved computational time is quite significant compared to the lookup table based approach even though the algorithmic approach is more versatile and capable of handling more cases.

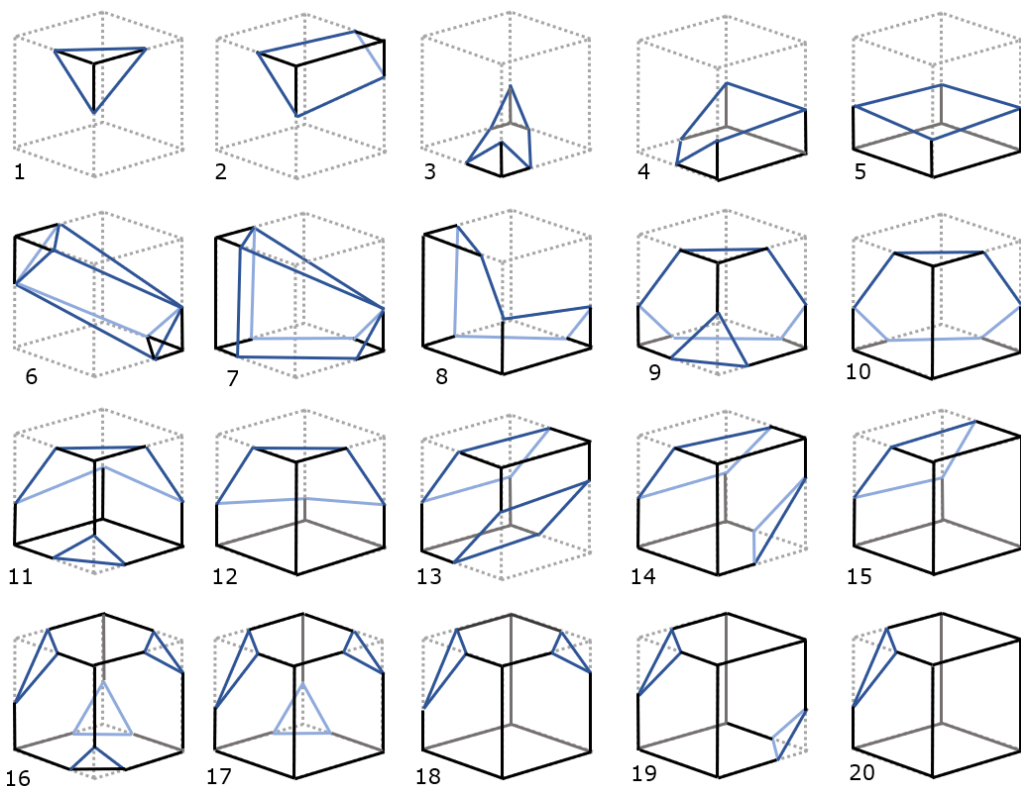


Figure 5: Set of 20 basic frame-sliced voxel configurations and their corresponding voxel shapes.

The lookup table-based approach is followed in this work to take advantage of its fast computational speed to generate a well-defined triangle mesh for the machined part geometry. A new lookup table is to be defined for all the possible frame-sliced voxel configurations. The 256 configurations counted in the previous section represent an elaborate set of possible frame-sliced voxel configurations with simplified frame edge configurations. After a detailed and extensive analysis of the 256 configurations, it has been observed that these configurations are in fact variants of just 22 basic configurations. All the 256 frame-sliced voxel configurations can be generated by rotational and mirror transformations of 22 basic configurations. Two of the 22 basic

configurations are simple ones: one is a null voxel with no active frame edge segments and the other is a full voxel with all the frame edges active. Figure 5 depicts the other 20 basic frame-sliced voxel configurations along with their corresponding frame-sliced voxel shapes. In the figure, the slice-front boundary is annotated with two shades of blue: dark blue for the directly visible boundary edges and lighter blue for the hidden edges of the slice-front boundary. This 22-case lookup table can be used to generate frame-sliced voxel shapes for all possible frame-sliced voxel configurations.

It should be noted that when constructing a frame-sliced voxel shape from a basic frame-sliced voxel configuration in the lookup table of Figure 5, connected face boundaries are preferred over disconnected face boundaries for the faces on the size sides of a given frame-sliced voxel. This preference is depicted in Figure 6 and made according to the most probable situation that occurs on machined workpiece surfaces. The selection is made because: (1) connected face boundaries promote feature preserving shapes; and (2) connected face shapes facilitate conformity between neighboring frame-sliced voxel shapes.

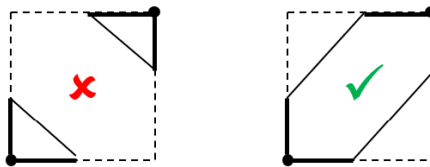


Figure 6: Connected face boundary preferred over disconnected face boundary.

4 APPLICABILITY VALIDATION

It can be demonstrated that the derived lookup table ensures the generation of a closed 2-manifold triangle mesh for all the simulated machined workpieces modeled as FSV-rep with simplified frame edge configurations. The demonstration is done in two steps. First, each frame-sliced voxel is to be verified to produce a 2-manifold triangle mesh patch for its sliced boundary surface. Second, the interfaces between neighboring frame-sliced voxels are all verified to have matching faces.

As can be observed in Figure 5, the sliced boundary surface for each of the basic frame-sliced voxel shapes clearly constitutes a 2-manifold triangle mesh patch. Recall that all the 256 possible frame-sliced voxel shapes are simply rotational and mirror transformations of the 22 basic shapes. Rotational transformations will not change the relative locations of the involved frame-crossing points. Hence, the resulting triangle mesh patch for the sliced boundary surface will remain 2-manifold. Similarly, mirror transformations retain the relative locations of the involved frame-crossing points with only the relative orientation of the points being opposite. It is, thus, assured that all of the 256 possible frame-sliced voxel shapes will have valid triangulation and produce 2-manifold triangle mesh patches for their sliced boundary surfaces.

The second step is to verify that matching situation exists between shared faces of neighboring frame-sliced voxels. Since the FSV-rep workpiece model is essentially a 26-separating voxel model, all the frame-sliced voxels definitely have a neighboring frame-sliced voxel across each of its faces containing a boundary edge of the triangle mesh patch associated with the frame-sliced voxel [5]. The matching is ensured by the set of 22 basic frame-sliced voxel shapes. The matching faces can be confirmed by considering all the possible cases for each face of a frame-sliced voxel. For each voxel face with four voxel corners, there are only 16 possible face configurations (or combinations of active corners with the simplified frame edge configurations) for a frame-sliced voxel as shown in Figure 7. Note that the faces of the basic frame-sliced voxel shapes as depicted in Figure 5 are all in one of the 16 face configurations in Figure 7. Since a voxel corner is either active for all of the voxels incident on it or inactive for all of them,

neighboring voxels always have the same set of active and inactive corners on their shared faces. Specifically, for a given face of a frame-sliced voxel, its configuration will conform to the configuration of the coincident face of the neighboring frame-sliced voxel according to the matching face configuration pairs listed in Table 1. This guarantees that the triangle mesh model generated from an FSV-rep workpiece model using the 22-case lookup table will always be closed and 2-manifold.

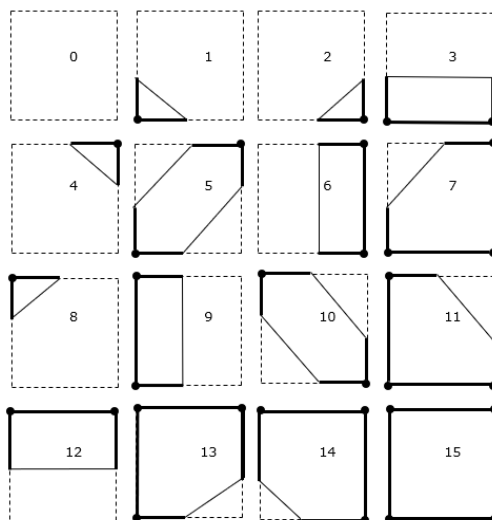


Figure 7: 16 possible face configurations for a frame-sliced voxel.

0 ⇔ 0
1 ⇔ 2; 1 ⇔ 8
2 ⇔ 4
3 ⇔ 3; 3 ⇔ 12
4 ⇔ 8
5 ⇔ 10
6 ⇔ 6; 6 ⇔ 9
7 ⇔ 11; 7 ⇔ 14
9 ⇔ 9
11 ⇔ 13
12 ⇔ 12
13 ⇔ 14
15 ⇔ 15

Table 1: Applicable matching face configuration pairs for shared faces of neighboring frame-sliced voxels.

It should be noted that the triangle mesh model generated from the simplified set of 22 basic frame-sliced voxels is valid if it can reasonably represent all the machined workpiece geometry that may arise. The main simplification applied in this work is to ignore the floating segment and isolated gap on the fine surface voxel frame edge. Ignoring a floating segment removes a sharp feature or pointed tip on the machined workpiece. Ignoring an isolated gap causes a shallow concave feature to disappear. All of these lost features are characterized with a dimension smaller than the fine voxel size. Since such features form only a very small fraction of the overall machined part surface, the impact on the resulting machined workpiece geometry is quite small and does not affect the global machined part geometry that needs to be preserved.

5 IMPLEMENTATION DETAILS

To generate a triangle mesh boundary surface for a voxel-based model such as the FSV-rep model using the per voxel processing approach, all the relevant information for every surface voxel needs to be stored and retrieved efficiently. To limit computer memory usage, a frame-sliced fine surface voxel in an FSV-rep model is stored only with the frame-crossing points that resides on its three primary voxel edges [5]. The frame-crossing points on the other 9 voxel edges are to be obtained from neighboring voxels to derive the shape of the frame-sliced voxel. Figure 8 depicts the 6 neighboring voxels to query for the frame-crossing point data. This means that the frame-crossing point data of a voxel are held by a group of maximum 7 voxels. To facilitate the retrieval of the frame-crossing point data, all the frame-sliced voxels in an FSV-rep model are stored in a binary search tree. More specifically, the binary search tree is to hold a sorted collection of frame-sliced voxels, each storing the frame-crossing points on its three primary edges. The binary search tree is structured such that to access the frame-crossing point data from the neighboring voxels of a given voxel can be done with $O(\log n)$ computational time cost where n is the number of all the frame-sliced voxels in the FSV-rep model held in the binary search tree. The computational time complexity is then $O(n \log n)$ for the whole FSV-rep model.

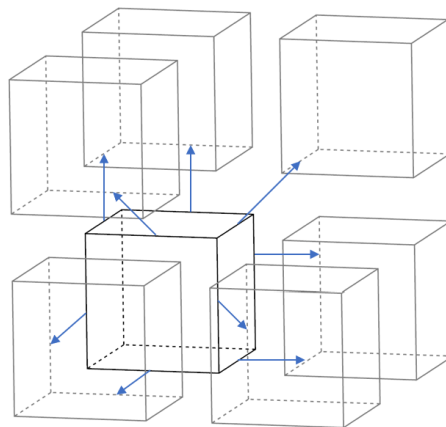


Figure 8: A frame-sliced voxel (bottom left at the back) with arrows to neighboring voxel edges to obtain frame-crossing points on non-primary edges.

With the frame-crossing point data on all of the 12 voxel frame edges available, the occupancy status for each of the 8 voxel corner points can be determined. As stated previously, a pair of frame-crossing points is to be stored on each of the three primary edges of a frame-sliced voxel. A corresponding pair of parameters $[u_1, u_2]$ is employed to hold the locations of the frame-crossing points on the line segment representing the voxel edge. If the machined surface normal at a frame-crossing point is along the positive direction of the primary edge, the resulting parameter

value is stored as u_2 . It should be noted here that the positive directions of the axes of the model coordinate system define the positive directions of the primary edges. If the machined surface normal is along the negative direction of the primary edge, the parameter value is stored as u_1 . Since the machined surface is generated by the cutting tool, its normal can be deduced as the reverse of the cutting tool surface normal at the point of interest. Such a scheme to store the frame-crossing point parameters leads to the straightforward deduction of the particular frame edge configuration as shown in Table 2 for a given parameter pair $[u_1, u_2]$ on a frame edge.

For example, if a surface with its normal along the positive direction of the primary edge intersects the primary edge at $u = 0.6$, then $u_2 = 0.6$. If another surface with its normal along the negative direction of the primary edge intersects the primary edge at $u = 0.4$, then $u_1 = 0.4$. If both intersections have happened, then $u_1 = 0.4$ and $u_2 = 0.6$, which will form a floating frame edge segment as shown in the first row of Table 2. This corresponds to the fact that the portion of the frame edge from the parameter value of 0.4 to 0.6 is part of the workpiece volume. If only $u_1 = 0.4$ is present and there is no intersection to set u_2 , we have a case as shown in the third row of Table 2 where an active frame edge attached to the corner point on the right side is formed. This corresponds to the fact that the corner point on the right is part of the workpiece volume whereas the corner point on the left is not.



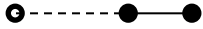
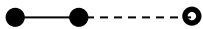
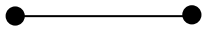
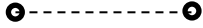
$\{u_1 \ \& \ u_2\} \in [0, 1)$ and $u_1 < u_2$	
$\{u_1 \ \& \ u_2\} \in [0, 1)$ and $u_1 > u_2$	
$u_1 \in [0, 1)$ and $0 > u_2$	
$u_2 \in [0, 1)$ and $0 > u_1$	
$\{u_1 \ \& \ u_2\} \geq 1$	
$\{u_1 \ \& \ u_2\} < 0$	

Table 2: Frame-crossing point parameter pairs and their corresponding frame edge configurations.

Using Table 2 to deduce the frame edge configuration for all the voxel frame edges, the voxel corner points to which active frame edge segments are attached are readily identified. Any corner point with an active frame edge segment attached is set to be active and any corner point with no active frame edge segment attached is set to be inactive. A frame edge is deemed completely active between two active corners and a frame edge is deemed completely inactive between two inactive corners. This means that the simplification employed in this work to ignore floating segments and isolated gaps on the voxel frame edges is done as the corner occupancy status is being determined. Once the corner occupancy status of a given frame-sliced voxel has been determined, the corresponding voxel shape can be obtained using the 22-case lookup table derived in this work with appropriate rotational and mirror transformations. For each item in the lookup table, the boundary loops of the involved slice-fronts as well as their triangulation have been predefined. Hence, after the occupancy status of all the corner points is known, the triangulation for all the surface patches within a frame-sliced voxel is readily done.

6 RESULTS AND COMPARISON

The applicability and computational advantages of the presented lookup table based method have been demonstrated via converting a number of typical machined parts in the FSV-rep model format to triangle mesh representations. The tested parts were selected with varying geometric complexity. The resulting triangle mesh representations of the machined part geometry are shown in Figure 9. It can be seen clearly in the figure that the geometry of these machined part has been captured well by the generated triangle meshes. It should be noted that due to the underlying grid-based modeling approach to approximate the part geometry, some detailed machined features cannot be represented exactly. In particular, the sharp machined edges have been approximated as chamfered edges. The sharp machined edges, however, can be easily restored from the chamfered edges using existing techniques such as that reported by Wang et al. [15].

To evaluate the gain in computational time from the use of the 22-case lookup table with reference to the existing algorithmic methods, quantitative comparisons have been made against the method developed by Ren et al. [14] which is applicable to simulated workpiece models in the tri-dexel as well as FSV-rep format. The algorithmic method has been known to generate valid 2-manifold triangle meshes for machined part geometry in most practical cases. As for computational efficiency, the test results shown in Table 3 clearly confirm the superiority of the 22-case lookup table against the algorithmic method by a factor of 2. This significantly reduces the time needed to generate a triangle mesh representation for the simulated machined workpiece and greatly facilitates time sensitive data processing tasks such as machining animation.

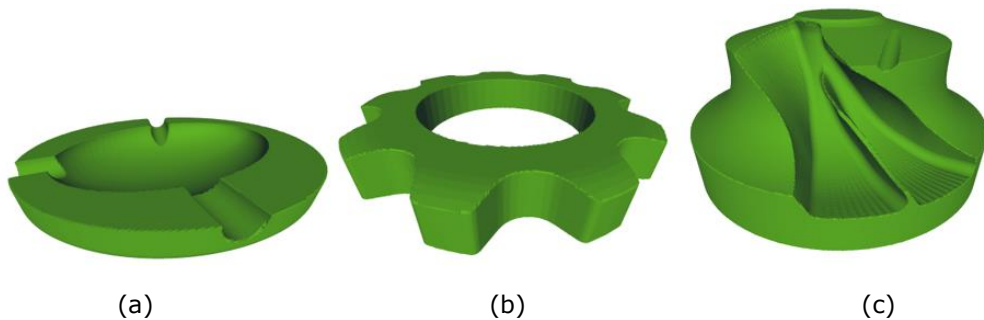


Figure 9: Triangle mesh representations generated in typical machining test cases: (a) Ash Tray; (b) Gear; and (c) Impeller Blade.

Test Case	No. of Vertices	No. of Triangles	Algorithmic Method (ms)	22-Case Lookup Table (ms)	Improvement Factor
Ash Tray	69,516	139,024	160	78	2.05
Gear	69,276	138,552	159	76	2.09
Impeller Blade	196,060	392,116	506	258	1.96

Table 3: Computational time comparison of triangle mesh generation by the presented 22-case lookup table-based method against an existing algorithmic method.

The models shown in Figure 9 clearly depict the varying level of complexity that can be handled by the FSV-rep model and the 22-case lookup table-based triangulation method. To further verify that the main assumptions taken during the lookup table derivation was appropriate for common machining cases, two more cases has been considered as shown in Figure 10.

Figure 10(a) shows a model with protuded features like the raised borders for the sides and extruded semi-cylinders extending out from the base. Figure 10(b) has a model which has small holes which have a diameter much smaller than the model size. The triangle mesh models generated from an FSV-rep model using the 22-case lookup table as shown in the figure have preserved the small features. This demonstrates that the assumptions made in Section 3.1 do not affect the validity of the models. The assumptions were made so that floating frame-edge segments could be ignored and the thin gaps of the frame edges could be closed. Ignoring the floating segments has not affected the ability of the model to preserve thin features such as the protusions in Figure 10(a). Closing the gaps has not removed the small shallow holes of the model in Figure 10(b), either.

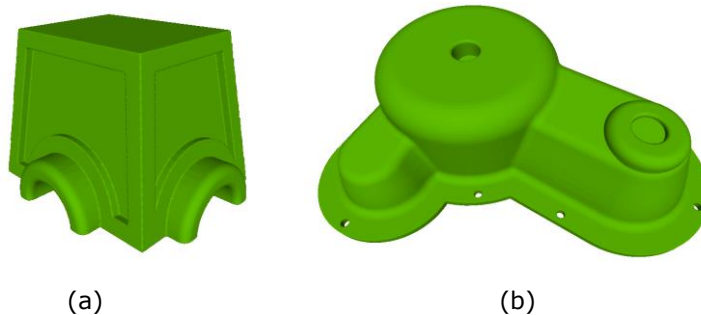


Figure 10: Triangle mesh representations generated in machining test cases: (a) with protruded features; (b) with small holes.

7 CONCLUSIONS

Computational time is one of the main concerns in the practical application of machining simulation technologies. In this paper, a lookup table of 22 pre-defined frame-sliced voxel shapes has been derived and used to quickly generate 2-manifold triangle mesh representations from simulated machined workpieces modeled in the FSV-rep format. The geometric simplifications made to the machined workpiece model in order to enable the use of a limited number of pre-defined frame-sliced voxel shapes, are seen to be justifiable. Specifically, no missing geometric details have been noted on the machined surfaces of the simulated workpieces for the involved test cases. For the small and thin machined features that would be missed due to the elimination of floating frame edge segments, it is also not an issue since the fine voxel size of the FSV-rep workpiece model should be set to be smaller than the expected thin feature size.

Jimin Joy, <https://orcid.org/0000-0003-4499-2366>

Jack Szu-Shen Chen, <http://orcid.org/0000-0002-1225-6340>

Hsi-Yung Feng, <http://orcid.org/0000-0001-6189-6910>

8 ACKNOWLEDGEMENTS

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the CANRIMT Strategic Network Grant as well as the Discovery Grant.

REFERENCES

- [1] Altintas, Y.; Kersting, P.; Biermann, D.; Budak, E.; Denkena, B.; Lazoglu, I.: Virtual process systems for part machining operations, *CIRP Annals - Manufacturing Technology*, 63(2), 2014, 585–605. <https://doi.org/10.1016/j.cirp.2014.05.007>

- [2] Chernyaev, E. V.: Marching cubes 33: Construction of topologically correct isosurfaces, CERN Report, CN/95-17, 1995.
- [3] Cohen-Or, D.; Kadosh, A.; Levin, D.; Yagel, R.: Smooth boundary surfaces from binary 3D datasets, Volume Graphics, Springer, London, 2000, 71–78. https://doi.org/10.1007/978-1-4471-0737-8_4
- [4] Ding, H.; Tang, J.; Zhong, J.; Wan, G.; Zhou, Z.: Simulation and optimization of computer numerical control-milling model for machining a spiral bevel gear with new tooth flank, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 230(10), 2016, 1897–1909. <https://doi.org/10.1177/0954405416640171>
- [5] Joy, J.; Feng, H. Y.: Frame-sliced voxel representation: An accurate and memory-efficient modeling method for workpiece geometry in machining simulation, Computer-Aided Design, 88, 2017, 1-13. <https://doi.org/10.1016/j.cad.2017.03.006>
- [6] Joy, J.; Feng, H. Y.: Efficient milling part geometry computation via three-step update of frame-sliced voxel representation workpiece model, International Journal of Advanced Manufacturing Technology, 92(5–8), 2017, 2365–2378. <https://doi.org/10.1007/s00170-017-0168-6>
- [7] Leu, M. C.; Peng, X.; Zhang, W.: Surface reconstruction for interactive modeling of freeform solids by virtual sculpting, CIRP Annals - Manufacturing Technology, 54(1), 2005, 131–134. [https://doi.org/10.1016/s0007-8506\(07\)60066-3](https://doi.org/10.1016/s0007-8506(07)60066-3)
- [8] Lewiner, T.; Lopes, H.; Vieira, A. W.; Tavares, G.: Efficient implementation of Marching Cubes' cases with topological guarantees, Journal of Graphics Tools, 8(2), 2003, 1–15. <https://doi.org/10.1080/10867651.2003.10487582>
- [9] Li, Y.; Lee, C. H.; Gao, J.: From computer-aided to intelligent machining: Recent advances in computer numerical control machining research, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 229(7), 2015, 1087–1103. <https://doi.org/10.1177/0954405414560622>
- [10] Lorensen, W. E.; Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm, Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87), 1987, 163–169. <https://doi.org/10.1145/37401.37422>
- [11] Oliver, J. H.; Goodman, E. D.: Direct dimensional NC verification, Computer-Aided Design, 22(1), 1990, 3–9. [https://doi.org/10.1016/0010-4485\(90\)90023-6](https://doi.org/10.1016/0010-4485(90)90023-6)
- [12] Peng, Q.; Loftus, M.: An image-based fast three-dimensional modelling method for virtual manufacturing, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 214(8), 2000, 709–721. <https://doi.org/10.1243/0954405001518080>
- [13] Pfister, H.; Hardenbergh, J.; Knittel, J.; Lauer, H.; Seiler, L.: The VolumePro real-time ray-casting system, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99), 1999, 251–260. <https://doi.org/10.1145/311535.311563>
- [14] Ren, Y.; Zhu, W.; Lee, Y. S.: Feature conservation and conversion of tri-dexel volumetric models to polyhedral surface models for product prototyping, Computer-Aided Design and Applications, 5(6), 2008, 932–941. <https://doi.org/10.3722/cadaps.2008.932-941>
- [15] Wang, Z.; Chen, J. S. S.; Joy, J.; Feng, H. Y.: Machined sharp edge restoration for triangle mesh workpiece models derived from grid-based machining simulation, Computer-Aided Design and Applications, 15(6), 2018, 905–915. <https://doi.org/10.1080/16864360.2018.1462571>
- [16] Zhang, W.; Peng, X.; Leu, M. C.; Zang, W.: A novel contour generation algorithm for surface reconstruction from dexel data, ASME Journal of Computing and Information Science in Engineering, 7(3), 2007, 203–210. <https://doi.org/10.1115/1.2752817>
- [17] Zhu, W.; Lee, Y.-S.: A visibility sphere marching algorithm of constructing polyhedral models for haptic sculpting and product prototyping, Robotics and Computer-Integrated Manufacturing, 21(1), 2005, 19–36. <https://doi.org/10.1016/j.rcim.2004.05.002>