ORIGINAL ARTICLE

Fast update of sliced voxel workpiece models using partitioned swept volumes of three-axis linear tool paths

Jimin Joy[1] · Hsi-Yung Feng[1]

Abstract

This paper presents a new and effective method to achieve fast machining simulation via the frame-sliced voxel representation (FSV-rep) workpiece modeling environment. The FSV-rep workpiece modeling scheme has been demonstrated as an efficient and accurate geometric modeling format for simulating general milling operations. The method presented in this paper specifically targets linear tool paths in three-axis milling. Based on the boundary representation of the linear tool swept volume and the voxel space for modeling the workpiece, it partitions the swept volume into a set of elemental regions, named as the swept volume regions (SVRs). With SVRs, the best performance of FSV-rep workpiece model update in three-axis milling is achieved. In the representative case studies, up to an order of magnitude faster computational performance has been observed compared to the generic approach of approximating the tool swept volume as a set of sampled tool instances. In addition, the quality of the simulated machined surface is much improved due to the use of non-approximated tool swept volumes. The efficient way to use SVRs in FSV-rep workpiece model update and the specific qualities of SVRs enabling the observed superior performance are also discussed in the paper.

Keywords Machining simulation · Computational efficiency · Three-axis milling · Linear tool path · Tool swept volume

1 Introduction

Geometric simulation is an integral part of virtual machining for the study and analysis of machining operations such as milling. The common goal is to validate the generated milling tool paths that have been optimized on the basis of various objectives [1–3] prior to physical machining. Milling operations are widely used for the production of parts in various industries. Effective visualization and computational validation of the machined part geometry are crucial to the support of various machining technology advancements such as tool path generation and optimization [4–7]. While multi-axis milling is employed to produce complex parts, three-axis milling that involves linear cutter motions with the cutter axis at a fixed orientation is the most common milling operation in practice. To have a geometric simulation system that offers fast and accurate simulation, a versatile workpiece model representation format together with an efficient

method to update the workpiece model is required. A suitable tool swept volume representation scheme that exploits the simplicity of three-axis milling tool paths to effectively update the workpiece model is also needed.

A versatile workpiece model representation format, named as frame-sliced voxel representation (FSV-rep), has been developed for general milling simulation [8]. The novel frame-sliced voxels on the workpiece surface enable sub-voxel modeling accuracy by using the voxel frame-crossing (FC) points to refine the surface voxels. The FC-points indicate the locations where the workpiece surface crosses the edge frame of the voxels representing the workpiece. Further, to ensure that the model is memory efficient, the voxel model is defined using two levels of coarse and fine voxels. The coarse level is used to model the bulk workpiece volume with a low memory requirement. Fine voxel grids are used within the coarse surface voxels to provide finer surface voxels to improve the workpiece model accuracy. With the FC-points, the frame-sliced voxel for each fine surface voxel provides the FSV-rep with the further improved accuracy to represent the machined workpiece geometry and the potential to update the workpiece model efficiently.

✉ Hsi-Yung Feng
feng@mech.ubc.ca

1   Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada

With an FSV-rep model constructed for the initial blank workpiece, an efficient three-step workpiece update process has been developed to compute the final machined part geometry for a given set of tool paths [9]. The three-step update process essentially utilizes the coarse–fine-frame levels of the FSV-rep model. First, in the coarse update, the bulk volume of the FSV-rep workpiece model affected by the tool paths is removed. Then, the fine voxels within the resulting coarse surface voxels are updated in the fine update step to refine the surface voxels to a finer resolution. Further, the edge frame of the resulting fine surface voxels is trimmed to attain the frame-sliced voxels with sub-voxel accuracy for the machined part geometry.

The FSV-rep workpiece model implemented with the three-step model update process yields the best combination of efficiency, accuracy and robustness for simulating general milling operations. Nonetheless, the linear tool paths used in three-axis milling result in tool swept volumes that are characterized by some geometric properties that can be exploited to develop effective workpiece volume removal tools to update the FSV-rep workpiece model even more efficiently and accurately. In this paper, we will identify a suitable representation of tool swept volumes resulting from linear tool paths in three-axis milling. The next section will review the existing tool swept volume representation methods and more importantly discusses the specific properties that can be exploited to update the FSV-rep workpiece model with further improved efficiency and accuracy.

2 Tool swept volume representation

Tool swept volume is the volume swept by the cutting tool for a single tool path or one NC block. When the tool paths are defined parametrically, they give the location and orientation of each tool instance as a function of a parameter varying across the tool path [10–13]. Analytically, the tool swept volume can then be expressed as a union of all tool instances along the tool path. Alternative parametric definitions of the tool swept volume as a two-parameter family of spheres [14] and Gauss curvature maps [15, 16] have also been reported. However, current parametric definitions of the tool swept volume are not directly compatible with the three-step update process of the FSV-rep workpiece model as they do not possess the explicit representation of the tool swept volume envelope surface required for efficient computation.

Boundary representation (B-rep) tool swept volumes that use connected surface elements to represent the swept volume envelope surface have also been introduced [17, 18]. B-rep tool swept volumes, and especially those defined using triangle meshes, can be utilized to update all types of workpiece representations. Notably, two-manifold triangle mesh

tool swept volumes can be generated even for the case of self-intersecting tool movements [19]. However, creation of such triangle mesh-based B-rep tool swept volumes is often a computationally demanding task and not suitable for efficient machining simulation. Nevertheless, the possibility to identify the boundary elements locally present in a coarse surface voxel is considered useful for the efficient update of FSV-rep workpiece models.

Approximating a tool swept volume as a collection of tool instances at different sampled locations along the tool path has been regarded as a practical approach useful for general milling simulation [19, 20]. Instead of updating the workpiece model with a single tool swept volume object, the sampled tool instances are applied one after another in this approach. Update of a workpiece model with each tool instance is fast and makes use of explicit closed-form solutions as the tool envelope surface of common milling cutters is composed of primitive surface elements [21, 22]. However, the sampling interval along a tool path will have to be small in order to limit the approximation error, which makes this approach computationally inefficient for linear tool paths in three-axis milling.

It is evident from the discussion above that the boundary representation of a tool swept volume yields an exact model of the swept volume with the notable capability to efficiently localize the update process. For linear tool paths in three-axis milling, the envelope surfaces of tool swept volumes consist of surface patches having closed-form solutions for intersecting the voxel frame edges. Hence, boundary representations of linear tool swept volumes can be directly used for FSV-rep workpiece update. For other nonlinear three-axis milling tool paths, it is also possible to update the FSV-rep workpiece model directly with the nonlinear tool swept volume if there exist closed-form solutions for intersecting the voxel frame edges with all the envelope surface patches. The fundamental workpiece model update process will be the same as that for linear tool swept volumes. Linear tool paths have been chosen as the primary focus of this paper due to their wide usage in three-axis milling.

3 Swept volume regions

The FSV-rep workpiece model update with respect to a given tool swept volume essentially involves two main operations: voxel classification and frame edge trimming. Using the tool swept volume boundary representation directly to update the FSV-rep workpiece model is not the best option. This is because the boundary representation of the tool swept volume is to be processed as a single entity, which will lead to sub-optimal and inefficient workpiece model update.

In general, a tool swept volume does not have an axis of symmetry. As a result, the voxel classification operation will

have to consider every part of the envelope surface of the tool swept volume. Also, for the subsequent trimming operation of the voxel frame edges, the various intersection points between the envelope surface and the voxel frame have to be determined. To minimize the number of workpiece voxels to be considered for the intersection computation, the common approach of utilizing the bounding box of the tool swept volume is not the best localization tool. For a linear tool swept volume in three-axis milling, its bounding box may be very large due to a long tool path and thus contains many voxels entirely outside the tool swept volume. This adversely affects the computational efficiency of the workpiece model update.

A preferred linear tool swept volume representation for fast FSV-rep workpiece model update is one that is much less intensive than a large set of sampled tool instances but is more effective in voxel localization than a single tool swept volume boundary representation. To achieve this, a new and effective tool swept volume representation has been developed, as presented in the following subsections.

3.1 Definition

A tool swept volume in general can be represented as a boundary representation using envelope surfaces defined by parametric expressions. Such a boundary representation typically includes three types of boundary elements: envelope surfaces, edges formed by the intersection of the envelope surfaces and vertices formed by the intersection of the boundary edges. For a generalized milling tool (in APT form) moving along a three-axis linear tool path, the numerous intersection calculations between the voxel frame edges of an FSV-rep workpiece model and all the boundary surfaces of the tool swept volume can be done without using an iterative solution technique. This is possible because four of the boundary surfaces are planar and four are conical, both having closed-form solutions for the intersection with the voxel frame edge. There are also two toroidal surfaces and two other surfaces formed by sweeping the toroidal section of the APT milling tool profile resulting in a quartic surface representation. Still, closed-form solutions are available for the intersection of these surfaces with a line segment [21, 23]. Hence, with the assurance that closed-form solutions exist for the intersection calculations for all the envelope surfaces of a linear tool swept volume B-rep, the focus is then to ensure that these calculations are done through effective localization. A set of swept volume regions (SVRs) obtained by partitioning the tool swept volume with respect to different boundary elements of the swept volume boundary representation are proposed in this work to achieve effective localization and defined as follows:
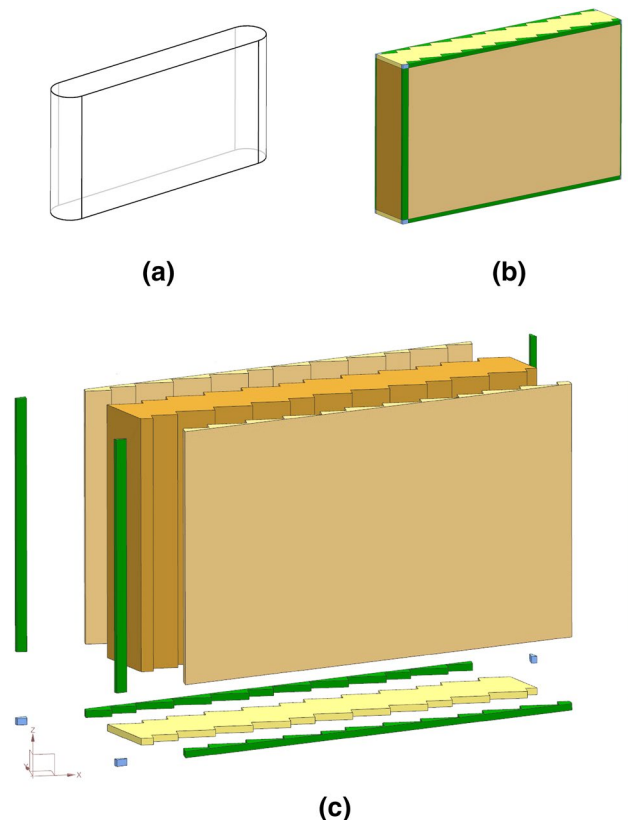
> Swept volume regions (SVRs) are the distinct portions of the tool swept volume overlapping with a minimum

possible set of voxels of the considered voxel space such that the same boundary elements of the tool swept volume are present in all of these voxels.

The SVRs as defined above offer the benefit of considering only a minimum number of boundary elements from the tool swept volume B-rep during the fine and frame updates of the FSV-rep workpiece model within each coarse surface voxel. Figure 1 shows the SVRs for a typical linear tool swept volume generated by a flat-end mill with the outer half of the tool instances at the two ends ignored for viewing simplicity. Once a voxel has been identified as associated with a particular SVR type, the boundary condition inside the voxel from the tool swept volume passing through it is fully known. This ensures that the calculations in the fine and frame updates are minimal in terms of the number of envelope boundary elements of the tool swept volume to be considered.

3.2 SVRs for a generalized end mill

This subsection outlines the SVRs for a generalized end mill. The generalized end mill is parametrically represented as

Fig. 1 Swept volume regions (SVRs): **a** typical linear tool swept volume from a flat-end mill; **b** partition of the tool swept volume into distinct regions; and **c** exploded view showing the partitioned SVRs

the APT form using seven parameters. A three-axis linear tool path is considered here with the associated tool swept volume B-rep shown in Fig. 2. It should be pointed out that SVRs for all other cases in three-axis milling will be a reduced version of this general case.

As shown in Fig. 2, there are 12 faces, 21 edges and 10 vertices for the generalized tool swept volume B-rep (excluding those forming the top cover, which is irrelevant for the workpiece model update). There is an SVR corresponding to each of these boundary elements, leading to a total of 43 boundary SVRs. The internal volume of the tool swept volume is an internal SVR. Hence, there are in total 44 SVRs for the general three-axis linear milling case. It has been assumed that the voxels of the voxel space are sufficiently small to keep the SVRs separate without some boundary elements falling into the same voxels causing some SVRs to merge together.

3.3 SVR types

The three SVR types and their distinguishing features in terms of the boundary condition within the associated voxels are given in this section. The SVRs are categorized according to the type of primary boundary element present in it. As a result, there are (1) face-based, (2) edge-based and (3) vertex-based SVRs. For face-based SVRs, there is just one surface element of the tool swept volume B-rep that passes through the constituent voxels. The surface element divides each of the face-based SVR voxels into two portions: one overlapping with the SVR and the other outside the SVR.

For edge-based SVRs, there are two surfaces incident on the edge passing through the constituent voxels. Thus, the portion of the edge-based SVR voxel completely overlapping with the SVR is determined by the intersection of

volume internal to both surfaces. Here, the internal volume is defined for a surface element as one from which the surface normal is pointing away. To perform fine and frame updates within an edge-based SVR, the specific surface to be considered to update a fine voxel or its frame should be first identified from the two surfaces incident on the edge. This can be done quite easily as the surface elements are always of a single value along the axial direction of the milling tool.

For vertex-based SVRs, there can be up to four surfaces incident on a vertex passing through the constituent voxel. Consequently, the update operation within a vertex-based SVR is the most involved. However, the general approach will be similar to that for an edge-based SVR. SVRs, thus, provide an effective way to update the workpiece model from a tool swept volume in a localized and generic manner.
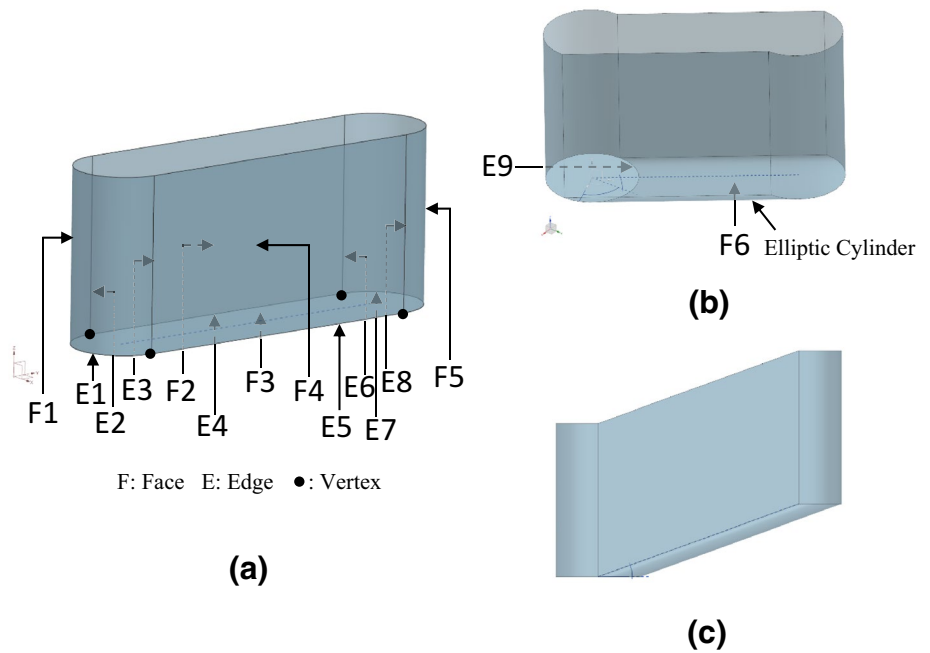
3.4 Simplified SVRs in actual milling operations

The SVRs developed for the generalized end mill correspond to the most complicated case to be handled. For commonly used end mills in practice such as flat-, ball-, tapered ball- and bull-end mills, the number of boundary elements in the associated tool swept volume is much smaller. Consequently, the number of SVRs needed to represent the tool swept volume is also much reduced. Figure 3 depicts the specific boundary elements for the associated SVRs for a flat-end mill following a linear tool path in three-axis milling. It can be seen that there are only five faces, eight edges and four vertices for the one-axis linear tool path. For the three-axis linear tool path, just one more face and one more edge are added. The added face is the elliptic cylinder face at the bottom, and the added edge is the boundary edge between the tool bottom face at the initial tool location and the elliptic

Fig. 2 Generalized tool swept volume B-rep and sample voxels overlapping with the three different SVR types

2. Edge-based SVR voxel

1. Face-based SVR voxel

3. Vertex-based SVR voxel
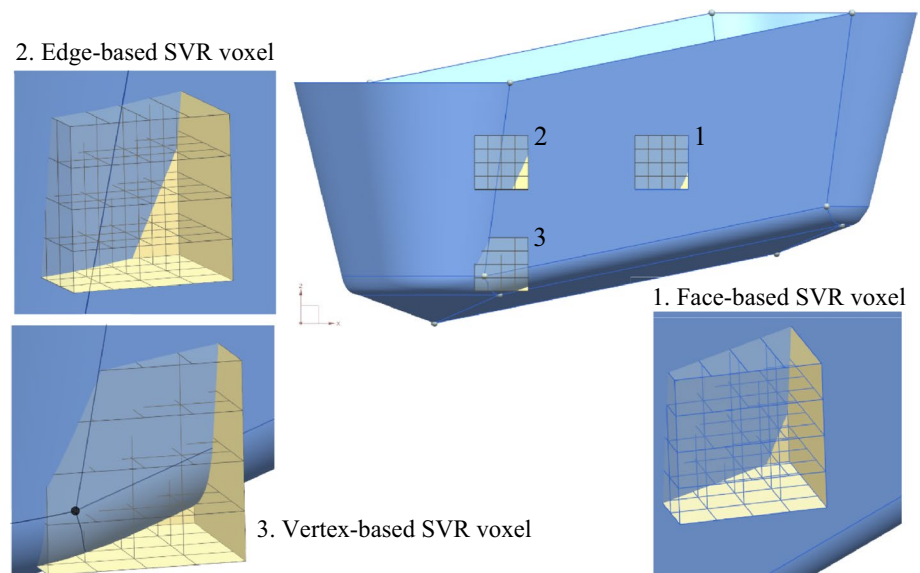
Fig. 3 Flat-end mill swept volume B-rep showing distinct boundary elements for the associated SVRs: **a** one-axis; **b** three-axis, angled view; and **c** three-axis, side view

cylinder face. Hence, it will be much easier to apply SVRs in actual milling operations.

4 FSV-rep workpiece model update with SVRs

The set of SVRs varies with the geometry of the associated tool swept volume and how the tool swept volume is positioned in the model coordinate system (MCS). As a result, to effectively use the category-specific SVRs for fast workpiece model update in three-axis milling, the linear tool paths need to be characterized first with reference to the MCS axes. The FSV-rep workpiece model then proceeds through the three-step coarse–fine-frame update process exploiting the fast computation potential of SVRs.

4.1 Tool path characterization

To facilitate coarse update computation on a workpiece model for a given three-axis linear tool path, it is necessary to distinguish and label the three MCS axes individually as the primary feed direction coordinate axis F_{dir}, lateral feed direction coordinate axis L_{dir} and tool axial direction coordinate axis A_{dir}. The MCS origin for the FSV-rep workpiece model is preferably set at the lower-left-back corner of the bounding box of the workpiece. This effectively ensures that all the tool paths transformed into the MCS are in the first octant. Further, for the purpose of assigning F_{dir}, L_{dir} and A_{dir}, whether the tool path is along the positive or negative direction of a particular MCS axis is not of importance. For a one-axis tool path, the

tool path follows along a particular MCS axis which becomes the F_{dir}. The MCS axis coinciding with the milling tool orientation is the A_{dir}, and the third/remaining MCS axis is the L_{dir}. For two-axis and three-axis linear tool paths, specifying an MCS axis to be the A_{dir} remains straightforward. However, the specification of F_{dir} and L_{dir} for the other two MCS axes needs some further considerations as outlined below.

For two-axis and three-axis linear tool paths, once A_{dir} is known and set as one of the MCS axes, the MCS plane containing F_{dir} and L_{dir} is readily known. The projection of a tool path curve onto this plane, tp_{proj}, will give the needed information to specify F_{dir} and L_{dir}. It is evident that tp_{proj} is identical to the tool path curve itself for two-axis tool paths not involving tool movement in A_{dir}. In this work, tp_{proj} is used to handle two-axis and three-axis tool paths in the same manner. With tp_{proj}, the MCS axis on the projection plane to which tp_{proj} is more inclined is specified as F_{dir} and the other MCS axis is then set as L_{dir}.

To apply the conditions stated above to label the MCS axes as A_{dir}, F_{dir} and L_{dir}, a four-component vector $[dX]$ of the tool path curve is formulated first:

$$[dX] = \left[dX_{C1}, dX_{C2}, dX_{A1}, dX_{A2} \right]. \tag{1}$$

For a tool path defined by two line segments, $C1(u)$ and $C2(u)$, $u \in [0, 1]$, as the trajectories of two specified points on the tool axis,

$$
\begin{aligned}
dX_{C1} &= X(C1(1)) - X(C1(0)) \\
dX_{C2} &= X(C2(1)) - X(C2(0)) \\
dX_{A1} &= X(C2(0)) - X(C1(0)) \\
dX_{A2} &= X(C2(1)) - X(C1(1)).
\end{aligned}
\tag{2}
$$

Then,

$$XLabel = \delta(dX_{C1}) \times 8 + \delta(dX_{C2}) \times 4 + \delta(dX_{A1}) \times 2 + \delta(dX_{A1}),$$
(3)

where $\delta(x) = 0$ if $x = 0$ and $\delta(x) = 1$ if $x \neq 0$. In the same way, YLabel and ZLabel can be evaluated from similar expressions of $[dY]$ and $[dZ]$. With the evaluated XLabel, YLabel and ZLabel values, A_{dir}, L_{dir} and F_{dir} can be assigned in sequence according to the guidelines below (which are synthesized from a series of valid combinations of XLabel, YLabel and ZLabel values):

1. The axis with a label value 3 or 15 is the A_{dir}.
2. If there is a label of value 0, the axis with the label value 0 is the L_{dir} and the remaining axis with a label value 12 is the F_{dir}.
3. If there are two labels of value 12, the one with a smaller first component in the four-component vector is the L_{dir} and the remaining axis is the F_{dir}.

4.2 Coarse update

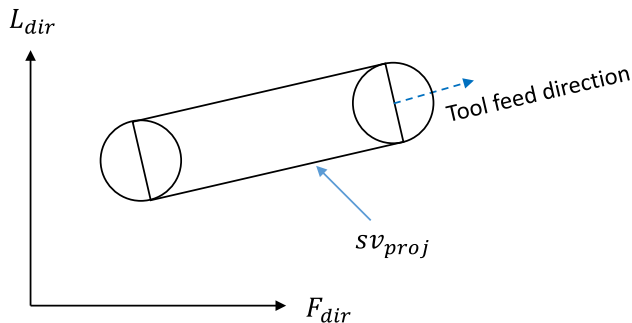In the coarse update step for the FSV-rep workpiece model, all the coarse surface voxels of the FSV-rep model that are

Fig. 4 Tool swept volume projection sv_{proj} on the base plane (perpendicular to A_{dir})

completely inside any tool swept volume are to be removed from the workpiece model. The new coarse surface voxels corresponding to the newly generated workpiece surface have to be identified as well. The underlying approach for the coarse update is similar to that of a voxelization scanning operation but without the need to scan the entire bounding box voxels of the tool swept volume. Still, it is applicable to all linear tool swept volumes with diverse boundary elements. Since the milling tool orientation is fixed, this allows the voxelization to be done locally using the tool swept volume projection sv_{proj} on the base plane perpendicular to the tool orientation (Fig. 4). Essentially, the method moves from using the MCS axis aligned bounding box to the boundary of sv_{proj}.

The boundary of sv_{proj} is associated with a set of voxels $\{V_{SVproj}\}$ covered by its extrusion along the tool orientation direction A_{dir}. The surface and (inner) volume voxels of the tool swept volume are a subset of $\{V_{SVproj}\}$. In order to avoid scanning through all of the voxels in $\{V_{SVproj}\}$, a simple 2D scan can be performed in the base plane containing the sv_{proj} and the vertical stack of voxels is handled the same way at each step along the 2D scan. This leads to a simple process as the linear tool swept volume has a fixed tool orientation and its projection on the base plane will consistently generate a well-defined sv_{proj} without a self-intersecting boundary.

The 2D scanning of shapes with a simple boundary is a widely studied problem with many effective solution techniques of which the sweep line algorithm is of special attraction due to its computational and data structural efficiency [24]. Hence, the coarse FSV-rep workpiece model update in this work extends the sweep line algorithm to a sweep plane voxelization algorithm for linear tool swept volumes. Algorithm 1 gives the fundamental principle for the sweep plane based coarse update. F_{dir} is taken as the primary sweeping direction and L_{dir} as the secondary or lateral sweeping direction for the simple scanning boundary of sv_{proj}.


Algorithm 1: Sweep plane based coarse update of FSV-rep workpiece models

For each step iF along F_{dir}
  For each step iL along L_{dir}
    $B \leftarrow$ base plane perpendicular to A_{dir}
    $\{L\} \leftarrow$ 4 grid lines along A_{dir} at locations $\{iF, iF+1\} \times \{iL, iL+1\}$
    $\{P\} \leftarrow$ intersection points of $\{L\}$ with tool swept volume bottom surface
    $V \leftarrow$ voxel containing the Max($\{P\}$)
    $Idx_A \leftarrow A_{dir}$ index of V voxel
    $Idx_{Atop} \leftarrow$ current A_{dir} index of top cover
    For all voxels with A_{dir} index from Idx_A+1 to Idx_{Atop}
      Delete from workpiece model
    End
  End
End

Fig. 5 Coarse update using SVRs: **a** sv$_{proj}$ boundary of a two-axis linear tool swept volume; **b** different sweeping sections according to sv$_{proj}$ boundary elements; and **c** inner coarse voxels removed

(a)   **(b)**   **(c)**
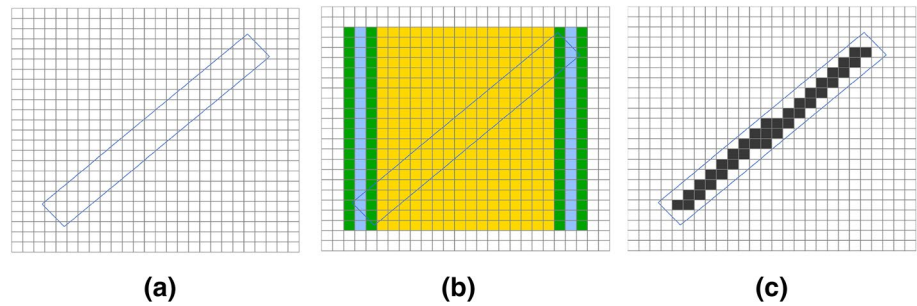
In the above algorithm, once the primary sweeping direction F_{dir} is identified, each tool swept volume will have different sweeping sections according to the varied boundary elements at the two ends in the lateral sweeping direction L_{dir}. For the two-axis linear tool path case shown in Fig. 5a, seven sweeping sections in different colors are depicted in Fig. 5b. The basic idea of scanning with the sweep plane is outlined as follows: (1) the scanning range is obtained in terms of the start and end voxel layer in the primary sweeping direction; (2) for each layer, the lateral sweeping bound is identified; (3) for each lateral sweeping step, the top and bottom voxels for the stack of tool swept volume voxels are identified; and (4) all voxels falling in the stacks are removed from the workpiece model. Figure 5c shows the coarse voxels removed in the voxel layer in the base plane.

4.3 Fine update

After the coarse update, the mapping of each SVR to the set of coarse surface voxels {CSV} of the FSV-rep workpiece model is established. An inversion of the mapping gives the CSV → {SVR} mapping. Here, the CSVs are the actual surface voxels with definite presence of the workpiece model envelope surface inside. The fine update for a CSV follows a similar scanning scheme as the coarse update

but within the bounds of the corresponding CSV subspace (Fig. 6). Essentially, the aim of the fine update for each CSV is to remove the fine voxels within the CSV that are completely within any of the SVRs passing through the CSV. The algorithm for the fine update is, thus, a modified version of Algorithm 1 for the coarse update but with the scanning step taken according to the fine resolution. Also, the fine update is performed for all SVRs present within a given coarse surface voxel.

It should be pointed out that in the fine update step, the active bottom surface element for a stack of fine voxels is the bottom bound for the SVR that is present across the particular fine voxel stack. For side face-based SVRs, the active bottom surface element can be the bottom voxel layer of the voxel subspace within the CSV as side faces are vertical for cylindrical end mills including the flat-, ball- and bull-end mills. Further, the update process can be made simpler depending on the SVR types. For face-based SVRs, there is only one tool swept volume surface element within the associated CSV. Thus, the fine voxel classification operation is quite simple due to the presence of only one surface element. For edge- and vertex-based SVRs, two and up to four surface elements may be present, respectively. The active bottom surface element has then to be identified for each stack of fine voxels based on the location of the stack with respect to

Fig. 6 Coarse surface voxels identified in Fig. 5 (left) and two sample coarse surface voxels after the fine update (right)
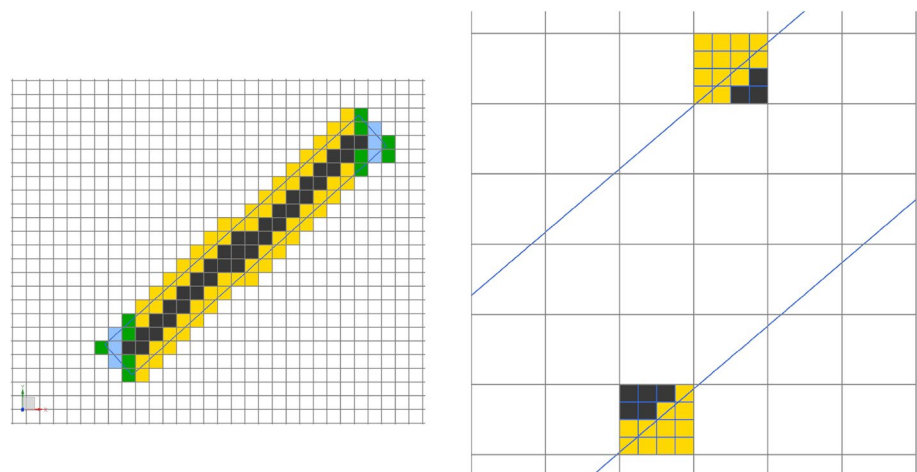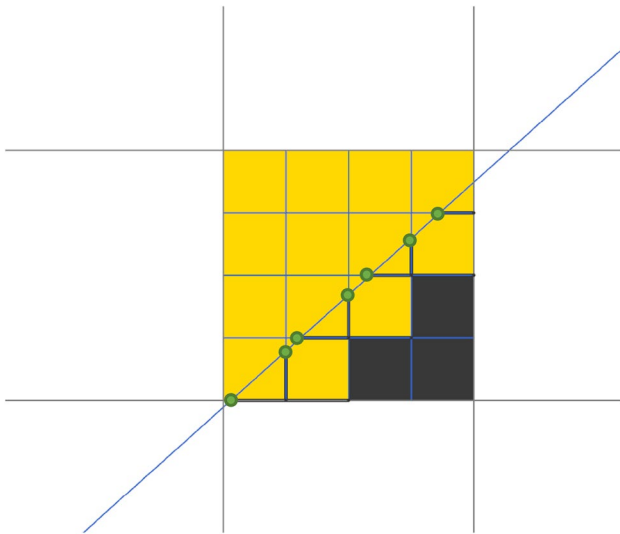
Fig. 7 Frame update performed on fine surface voxels within a CSV from a face-based SVR (A_{dir} view)

the projection of the tool swept volume edge elements on the bottom voxel layer of the CSV voxel subspace.

4.4 Frame update

The objective of the frame update is to refine the fine surface voxels. A typical example is shown in Fig. 7. The frame update performs the intersection of the fine voxel edge frames with the associated SVRs. The available CSV → {SVR} mapping, used previously for the fine update, is again utilized. After the fine update, all the fine voxels within each CSV have been labeled with a correct status of inner, surface or outer/removed. Performing the frame update only after the fine update ensures that a fine voxel that is a surface voxel with respect to a particular SVR will not be frame updated if it has been removed by another SVR within the CSV during the fine update. This reduces the computational load as fine update only requires intersection of the fine voxel frame edge along the A_{dir} as discussed in the previous subsection, whereas frame update for a particular surface fine voxel will require intersection of all the relevant voxel frame edges with the SVR boundary envelope. Algorithm 2 gives the general frame update steps for the various SVR types.


Algorithm 2: Frame update within coarse surface voxels

For each *SVR* present in a *CSV*
    For each step *iF* along F_{dir}
        For each step *iL* along L_{dir}
            B ← base plane perpendicular to A_{dir}
            {L} ← 4 grid lines along A_{dir} at locations $\{iF, iF{+}1\} \times \{iL, iL{+}1\}$
            $Surf_{btm}$ ← active bottom surface elements of the *SVR* for **{L}**
            {P} ← intersection points of **{L}** with $Surf_{btm}$
            {V} ← surface voxels from voxels for min(**{P}**) to max(**{P}**)
            For each voxel *V* in **{V}**
                If *V* is not active
                    Skip to next *V*
                Update FC-points for the edge frame of *V*
            End
        End
    End
End

(a)

(b)

(c)

(d)
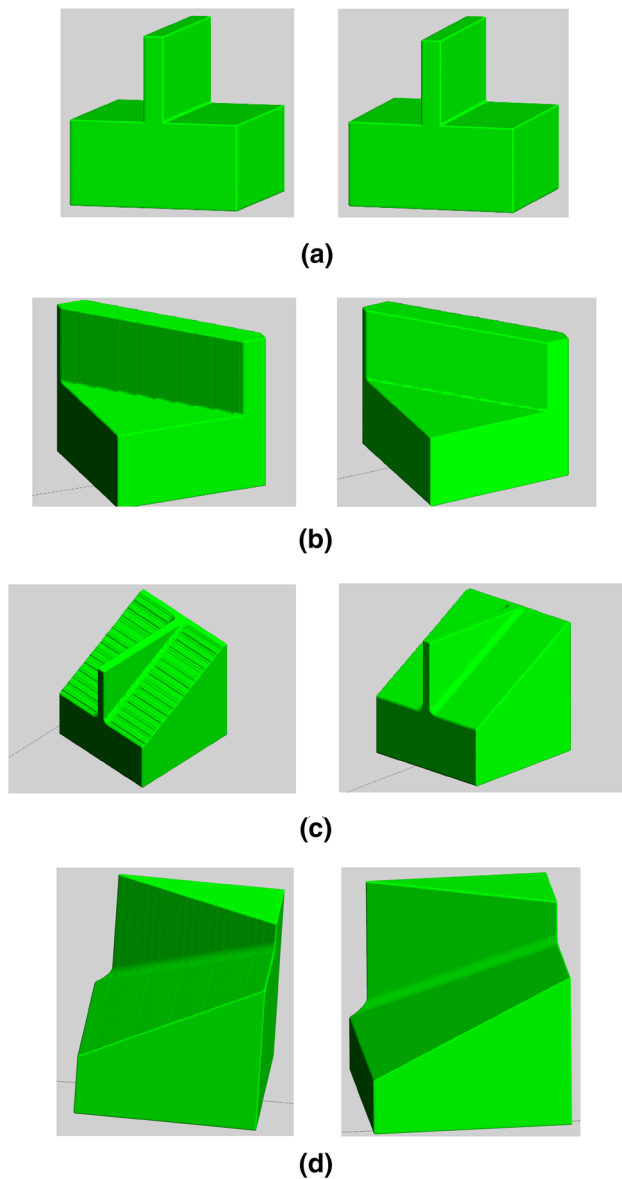
Fig. 8 Comparison of surface quality of simulated workpiece models using sampled tool instances (left) and SVRs (right): **a** case 1A; **b** case 2A; **c** case 2AV; and **d** case 3A

5 Implementation specifics

A machining simulation tool for linear tool paths in three-axis milling, using FSV-rep for workpiece model representation and SVRs for workpiece model update, has been developed. A Windows laptop with a two-core CPU @ 2.50 GHz and 8 GB of primary memory was used to perform the case studies. As the first step of a case study, an FSV-rep workpiece model is derived for the initial blank workpiece. With the initial FSV-rep workpiece model created, the linear milling tool paths are characterized according to the procedure presented in Sect. 4.1. This is done to ensure that

the SVRs-based workpiece model update is applied appropriately and only for the corresponding linear tool paths. For a particular linear tool path, once Algorithm 1 has been executed to remove the inner coarse voxels from the FSV-rep workpiece model, a preparatory task is performed prior to the fine update operation. This intermediate task is to create the SVR → {CSV} mapping. With the inverted CSV → {SVR} mapping, the fine and frame update operations, as presented in Sects. 4.3 and 4.4, are then done. In this work, each SVR type is implemented to have its own fine and frame update operation definitions. This enables the corresponding workpiece model update operations to be automatically invoked according to the SVR types present in each CSV. It should be noted that even though there are specific fine and frame update definitions for each SVR type, they all follow the general workflow presented in Sects. 4.3 and 4.4. The individual definitions are utilized simply to accommodate different number and types of boundary elements that constitute each SVR type.

6 Case studies and discussion

To demonstrate the benefits of SVRs in workpiece model update for linear tool paths in three-axis milling, different case studies with increasing complexity have been performed. In each case study, the performance of SVRs was compared with that of an existing approach which approximates the tool swept volume as a set of sampled tool instances [9]. As shown in Fig. 8, Case 1A is to machine an inverted T-section part. This case involves only one-axis

Table 1 Simulation time comparison of FSV-rep workpiece model update with sampled tool instances and SVRs for the cases illustrated in Fig. 8

Case	Sampled instances (ms)	SVRs (ms)	Improvement ratio
1A	398	15	26.5
2A	421	22	19.1
2AV	200	25	8.0
3A	203	40	5.0

Table 2 Simulation time comparison of tri-dexels with tool swept volumes and FSV-rep with SVRs for the cases illustrated in Fig. 8

Case	Tri-dexels with tool swept volumes (ms)	FSV-rep with SVRs (ms)	Improvement ratio
1A	41	15	2.73
2A	47	22	2.14
2AV	11	25	0.44
3A	260	40	6.50

linear tool paths of a flat-end mill and is, thus, the first and simplest case study. Case 2A is a rotated case of case 1A, and all the linear tool paths, thus, involve two-axis tool motions (although both tool motions are lateral with respect to the tool axis). Case 2AV again involves two-axis motions of a flat-end mill, but in this case, a tool motion component along the vertical tool axis is present. Such a vertical tool motion alters the bottom envelope surface of the tool swept volume to the surface of an elliptic cylinder and causes the associated SVRs to contain much more complex boundary elements. Case 3A involves a ball-end mill moving along three-axis linear tool paths and results in the most complex operations for updating the workpiece model among the four cases.

It is evident from the simulated workpiece model geometry shown in Fig. 8 that the false scallops that are visually noticeable from the sampled tool instances approach no longer exist for the SVRs approach. For case 1A, the improvement is not noticeable because the planar side surfaces are aligned with a voxel grid axis. Noticeable improvements are seen for the machined side surfaces in cases 2A and 3A and for the bottom machined surface in case 2AV. It is also worth noting that the bottom machined surface in case 3A with sampled tool instances also yields visually acceptable quality. This is because the spherical bottom of a ball-end mill produces smaller steps between sampled tool instances than the planar bottom of a flat-end mill. Nevertheless, there is still a recognizable improvement for that machined surface with SVRs.

Comparison of the computational performance in terms of the simulation time is listed in Table 1 to further showcase the advantages of SVRs. It can be seen that the computational speed with SVRs increases by an order of magnitude for the one-axis and two-axis machining cases. It is also noted that the performance improvement increases from case 3A toward case 1A. This is expected since the involved tool swept volume envelope surfaces and SVRs boundary elements become less geometrically complex from case 3A toward case 1A.

Table 2 shows the simulation time comparison of the update of an FSV-rep workpiece model with SVRs against the update of a tri-dexel workpiece model with tool swept volumes [25]. The FSV-rep update is faster than the tri-dexel update in all of the cases except for case 2AV. The faster performance of FSV-rep is attributable to the minimization of intersection calculations due to the two levels of voxel model update prior to the frame update step. For tri-dexels, however, the intersection calculations have to be performed for each dexel crossing the tool swept volume of a tool path. The tri-dexel model has to be updated with each individual tool path to compute the final machined part geometry. For cases 1A, 2A and 3A, multiple levels of tool paths at different vertical heights were employed to mimic a constant
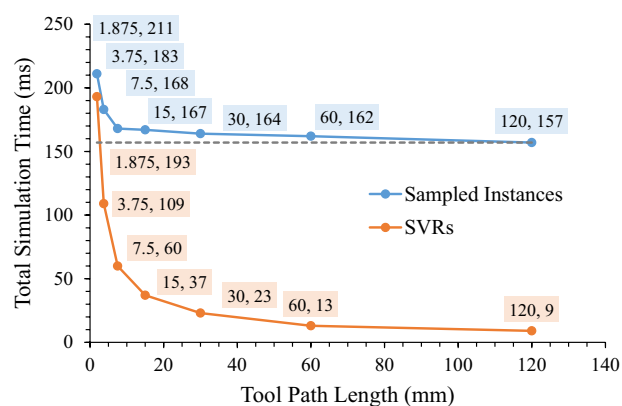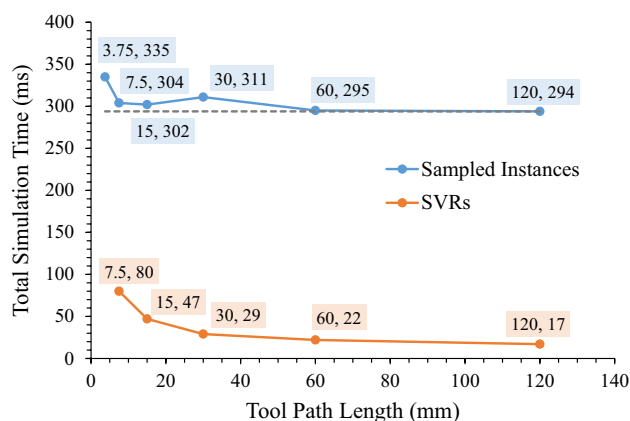
Fig. 9 Comparison of total simulation time using sampled tool instances and SVRs for varying tool path segment length for a 120 mm: **a** one-axis linear tool path and **b** two-axis horizontal linear tool path

axial depth of cut milling scenario in practical machining. Such overlapping milling tool paths lead to a larger number of intersection calculations and thus longer computational time for the tri-dexel model. For case 2AV, a simplified, single level of tool paths was used to machine the workpiece in one step. In the absence of the overlapping tool paths, the tri-dexel update is seen to be faster.

Another important aspect to evaluate is the effect of individual tool path segment length or tool swept volume size on the resulting simulation time using SVRs. Two specific machining cases were devised for this evaluation. A single one-axis linear tool path and a single two-axis horizontal linear tool path both of 120 mm long were simulated with successively reduced tool path segment length by bisecting all the tool path segments in the immediate last simulation. All such simulated milling operations have the same amount of machining work, but each has twice the number of (shorter) tool path segments in comparison with its predecessor. As shown in Fig. 9, the SVRs approach outperforms the

sampled tool instances approach for both the one-axis and two-axis cases even when the tool path segments become very short. For the sampled tool instances approach, if the sampling is done for the entire 120-mm linear tool path, the total simulation time will be ideal and remain constant for all the simulated operations, as indicated by a dashed gray line in the figure charts. In practice, however, each tool path segment should be sampled individually as neighboring tool path segments almost always follow difference routes. The individual tool path sampling is subject to inefficient coarse update when the tool path segment length gets close to the coarse voxel grid spacing. This causes the sampled tool instances approach to increase its simulation time at small tool path lengths. For the SVRs approach, its simulation time also increases with reduced tool path segment length as seen in the figure chart. This means that SVRs are most computationally favorable for longer linear tool paths.

The advantages of the SVRs approach are limited by the individual tool path segment length according to a particular threshold. This threshold varies for each tool type and reduces from case 1A toward case 3A (Fig. 8). As shown in Fig. 9, a linear tool path of 1.875 mm length can be simulated for the one-axis milling case but cannot be done for the two-axis horizontal milling case. More specifically, the method to obtain the $SVR \rightarrow \{CSV\}$ mapping is applicable only when the tool path length is large enough to avoid the boundary elements at the two extreme ends along F_{dir} from occurring in the same L_{dir} sweeping section. Also, for short tool paths, some SVR types can vanish and the SVRs approach will require more rigorous voxel identification techniques or may not be feasible at all. Nevertheless, the observed threshold in tool path segment length is sufficiently small, enabling SVRs to advantageously handle a wide range of three-axis linear milling operations.

7 Conclusions

This paper has presented a fast FSV-rep workpiece model update method using swept volume regions (SVRs) as a partitioned tool swept volume representation for three-axis linear tool paths. The superior performance of SVRs can be attributed to its abilities to: (1) directly identify the voxels that need to be updated and (2) perform the involved intersection calculations via formulations with closed-form solutions. Three-axis curved tool paths such as circular, helical and free-form tool paths are not part of the present work. It is possible to implement the SVRs approach for the circular and helical tool paths. However, due to the complexity of the associated geometry, there is only modest improvement in computational speed over the existing sampled tool instances approach. Since circular and helical tool paths are much less common than linear tool paths in machining a prismatic

mechanical part, the potential contribution to the reduction in the overall simulation time is quite low by implementing the SVRs approach. Free-form curved tool paths cannot be used directly with SVRs because the resulting tool swept volumes are not confined by quartic surface representations. These tool paths can be approximated as piecewise linear tool paths. Unfortunately, the approximated linear tool path segments are often shorter than the required tool path length threshold, rendering the SVRs approach inapplicable.

References

1. Kersting P, Zabel A (2009) Optimizing NC-tool paths for simultaneous five-axis milling based on multi-population multi-objective evolutionary algorithms. Adv Eng Softw 40(6):452–463
2. Manav C, Bank HS, Lazoglu I (2013) Intelligent toolpath selection via multi-criteria optimization in complex sculptured surface milling. J Intell Manuf 24(2):349–355
3. Fountas NA, Vaxevanidis NM, Stergiou CI, Benhadj-Djilali R (2017) A virus-evolutionary multi-objective intelligent tool path optimization methodology for 5-axis sculptured surface CNC machining. Eng Comput 33(3):375–391
4. Li DY, Peng YH, Yin ZW (2006) Interference detection for direct tool path generation from measured data points. Eng Comput 22(1):25–31
5. Chen ZC, Fu Q (2014) An efficient, accurate approach to medial axis transforms of pockets with closed free-form boundaries. Eng Comput 30(1):111–123
6. Fountas NA, Benhadj-Djilali R, Stergiou CI, Vaxevanidis NM (2017) An integrated framework for optimizing sculptured surface CNC tool paths based on direct software object evaluation and viral intelligence. J Intell Manuf 30(4):1581–1599
7. Fountas NA, Vaxevanidis NM, Stergiou CI, Benhadj-Djilali R (2018) Globally optimal tool paths for sculptured surfaces with emphasis to machining error and cutting posture smoothness. Int J Prod Res. https://doi.org/10.1080/00207543.2018.1530468
8. Joy J, Feng HY (2017) Frame-sliced voxel representation: an accurate and memory-efficient modeling method for workpiece geometry in machining simulation. Comput Aided Des 88:1–13
9. Joy J, Feng HY (2017) Efficient milling part geometry computation via three-step update of frame-sliced voxel representation workpiece model. Int J Adv Manuf Technol 92(5–8):2365–2378
10. Fleisig RV, Spence AD (2001) A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining. Comput Aided Des 33(1):1–15
11. Langeron JM, Duc E, Lartigue C, Bourdet P (2004) A new format for 5-axis tool path computation, using Bspline curves. Comput Aided Des 36(12):1219–1229
12. Sullivan A, Erdim H, Perry RN, Frisken SF (2012) High accuracy NC milling simulation using composite adaptively sampled distance fields. Comput Aided Des 44(6):522–536
13. Sarkar S, Dey PP (2015) Tolerance constraint CNC tool path modeling for discretely parameterized trimmed surfaces. Eng Comput 31(4):763–773
14. Aras E, Feng HY (2011) Vector model-based workpiece update in multi-axis milling by moving surface of revolution. Int J Adv Manuf Technol 52(9–12):913–927

15. Lee SW, Nestler A (2011) Complete swept volume generation, Part I: swept volume of a piecewise C1-continuous cutter at five-axis milling via Gauss map. Comput Aided Des 43(4):427–441

16. Lee SW, Nestler A (2011) Complete swept volume generation—part II: NC simulation of self-penetration via comprehensive analysis of envelope profiles. Comput Aided Des 43(4):442–456

17. Weinert K, Du S, Damm P, Stautner M (2004) Swept volume generation for the simulation of machining processes. Int J Mach Tools Manuf 44(6):617–628

18. Yang Y, Zhang W, Wan M, Ma Y (2013) A solid trimming method to extract cutter-workpiece engagement maps for multi-axis milling. Int J Adv Manuf Technol 68(9):2801–2813

19. Gong X, Feng HY (2016) Cutter-workpiece engagement determination for general milling using triangle mesh modeling. J Comput Des Eng 3(2):151–160

20. Ferry W, Yip-Hoi D (2008) Cutter-workpiece engagement calculations by parallel slicing for five-axis flank milling of jet engine impellers. ASME J Manuf Sci Eng 130(5):051011

21. Chung YC, Park JW, Shin H, Choi BK (1998) Modeling the surface swept by a generalized cutter for NC verification. Comput Aided Des 30(8):587–594

22. Altintas Y, Engin S (2001) Generalized modeling of mechanics and dynamics of milling cutters. CIRP Ann Manuf Technol 50(1):25–30

23. Shmakov SL (2011) A universal method of solving quartic equations. Int J Pure Appl Math 71(2):251–259

24. Fortune S (1987) A sweepline algorithm for Voronoi diagrams. Algorithmica 2:153–174

25. Lee SW, Nestler A (2012) Virtual workpiece: workpiece representation for material removal process. Int J Adv Manuf Technol 58(5–8):443–463