Title:
Visualization of 3+2 Axis Milling Result by Combining Multiple Z-map Models

Authors:
Qi Chen, 20nm470t@vc.ibaraki.ac.jp, Ibaraki University
Masatomo Inui, masatomo.inui.az@vc.ibaraki.ac.jp, Ibaraki University

Keywords:
NC Milling Simulation, Dexel Modeling, Solid Model Conversion

DOI: 10.14733/cadconfP.2021.88-92

Introduction:
Geometric NC milling simulations are generally performed before the actual cutting to visualize the machining result and detect potential errors such as cutter gouges in the machining process. The milling simulation method based on the Z-map representation of the workpiece shape is most commonly used because of its implementation ease, speed, and robustness. In this method, the workpiece shape is represented as a collection of vertical line segments, which are extended from each grid point of a square mesh placed on a horizontal reference plane (Fig. 1(a)). For each linear motion of the cutter along the path, the intersection between the segments and the cutter swept volume is computed and removed from the workpiece model. To represent the machining results accurately, the reference plane for the Z-map must be set perpendicular to the spindle direction of the cutter. Therefore, machining simulation using the Z-map is basically performed only for 3-axis machining, in which the spindle direction is fixed in the z-axis direction.

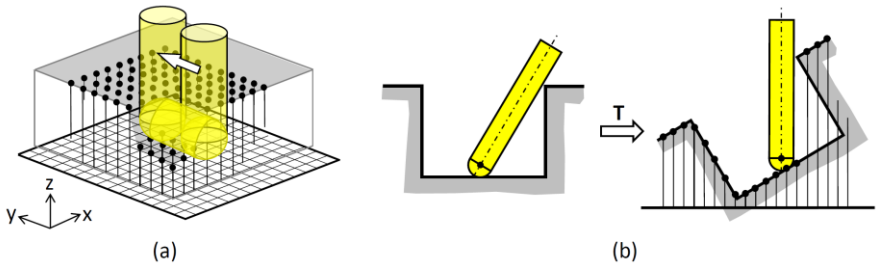

Fig.1: (a) Z-map based milling simulation. (b) 3+2 axis machining simulation using Z-map.

Recently, the use of 3+2 axis machining, in which machining is performed by tilting the direction of the tool spindle, has increased in the mold and die fabrication. Using this method, both the machining accuracy and machining cost can be simultaneously improved. When the direction of the spindle is limited to one direction, 3+2 axis machining can be geometrically simulated using a Z-map by applying a coordinate transformation **T** to the workpiece shape and the cutter path so that the spindle direction of the cutter is parallel to the z-axis (Fig. 1(b)). After the simulation, the resultant shape is transformed back to the original coordinate frame using $\mathbf{T}^{-1}$. When machining a complex mold, the milling operation can be performed by switching multiple spindle directions, which cannot be handled by a single Z-map model. Therefore, in the geometric simulation of the 3+2 axis machining, a time-consuming simulation technique using B-reps, voxels, or triple-dexels is generally used.

We propose a novel method to visualize the result of the 3+2 axis machining operations. The proposed method decomposes a complex 3+2 axis machining operation into a set of single-spindle-direction-machining operations. Each machining operation is simulated using a Z-map. Subsequently, all Z-map models representing the machining results for each spindle direction are combined into one solid model, which represents the total machining result. In our software, a triple-dexel model [1] is used to represent the resultant shape. This simulation method can correctly visualize the 3+2 axis machining result at a high speed using the rapidity of the Z-map based milling simulation and the expressiveness of the triple-dexel model. The effectiveness of the technology was verified using software and by performing computational experiments.

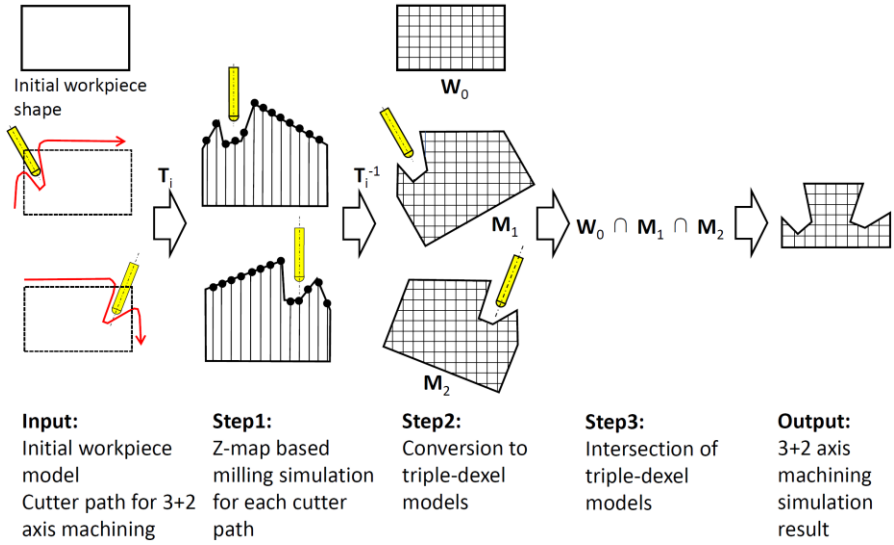

Fig. 2: Outline of our algorithm for simulating 3+2 axis machining with multiple Z-maps.

Basic Idea:

Fig. 2 illustrates the outline of the algorithm. The input data of the algorithm consist of a polyhedral STL model representing the initial shape of the workpiece and a set of cutter path data and their corresponding cutter data for 3+2 axis machining. The data of each cutter path are defined for one cutter with a specific spindle direction. The output of the algorithm is a solid model representing the resultant shape after milling. In our implementation, the triple-dexel model is used for representing the resultant shape.

Dexel modeling is known as an extension of the Z-map. In this method, the object shape is represented using a bundle of z-axis-aligned segments defined for each grid point of a square mesh in the xy plane. With dexel modeling, near-vertical surfaces are prone to inevitable large shape errors caused by a finite grid resolution. The triple-dexel model was proposed to overcome this non-uniformity of the representation accuracy. In this representation, the solid shape is not only defined by a z-axis-aligned dexel model but also by an x-axis-aligned dexel model based on a square mesh in the yz plane and a y-axis-aligned dexel model based on a mesh in the zx plane (Fig. 3). Because a triple-dexel model defined based on properly aligned square meshes in the xy, yz, and zx planes is equivalent to a voxel model, it can be easily converted to a polyhedral model using boundary evaluation techniques such as Marching Cubes algorithm.

Consider a single 3+2 axis machining operation with a cutter in a certain spindle direction. By performing a proper coordinate transformation $\mathbf{T}_i$ on the initial workpiece model and each cutter path, and by executing Z-map based milling simulation with the transformed data, a machined shape in the Z-map representation is obtained, as mentioned in Step1 in Fig. 2. The machined shape is converted to an equivalent polyhedral model by using a method explained later. The polyhedral model is transformed into the original coordinate frame by applying $\mathbf{T}_i^{-1}$ on it. The model is then converted to a

triple-dexel model (Step2). When 3+2 axis machining is performed for multiple spindle directions, the same number of machined shapes can be obtained by repeating the above-described processing for each spindle direction. A resultant workpiece model reflecting all 3+2 machining operations is finally obtained by combining multiple machined shapes using the Boolean intersection operation (Step3). In this method, the simulation time can be reduced significantly, because a simple conversion of the polyhedral model into a triple-dexel model and several Boolean intersection operations of dexel models are necessary only after high-speed milling simulation using the Z-map.

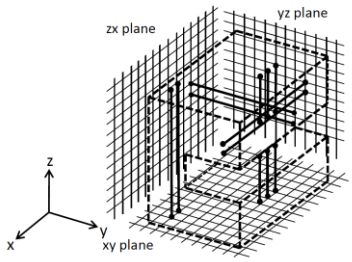

Fig. 3: Triple-dexel model.

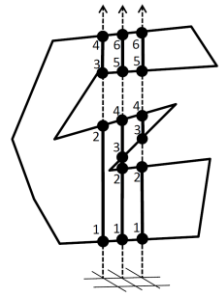

Fig. 4: Ray algorithm for polyhedron-to-dexel conversion.

Details of Algorithm:

Conversion from Milling Result Z-map to Triple Dexel Model

To realize the simulation method, the following three steps are necessary:

- Z-map based milling simulation (Step1).
- Conversion from a Z-map model to a triple-dexel model (Step2).
- Boolean intersection computation of triple-dexel models (Step3).

We used our developed Z-map based milling simulation software in this study [2]. This system can visualize the milling result in a short time using the depth buffer function of graphics processing unit (GPU). Because the calculation of the Boolean intersection of two triple-dexel models can be decomposed to a set of simple Boolean intersection calculations of segments on the same line, its implementation is straightforward. Therefore, in the following, only the method for converting the machined shape model in the Z-map representation to a triple-dexel model is explained.

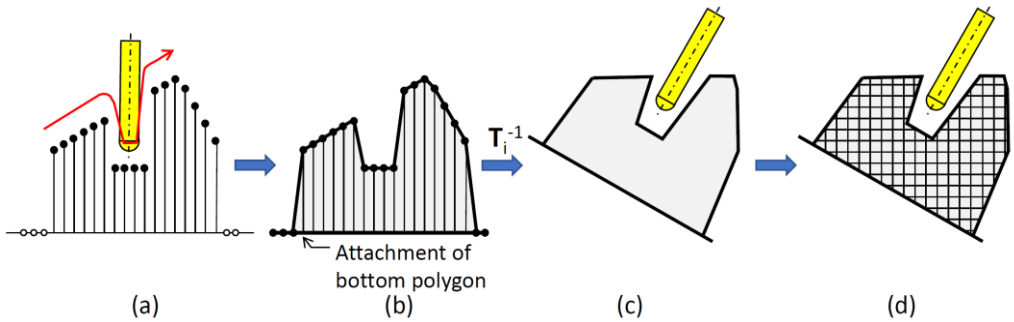

Fig. 5: (a) Z-map based milling simulation result. (b) Conversion to closed polyhedral model. (c) Transformation to original coordinate frame. (d) Conversion to triple-dexel model using ray algorithm.

A Z-map model can be recognized as a set of vertical line segments. It is difficult to convert such a shape representation directly into a triple-dexel model. Therefore, we first convert a Z-map model into a closed polyhedron. Then we further convert it into a triple-dexel model using the ray algorithm. Before the conversion to the triple-dexel model, a coordinate transformation $\mathbf{T}_i^{-1}$ is applied to move the polyhedron to the original coordinate frame. In the ray algorithm, a square mesh in the xy plane is

prepared. Corresponding to each grid point, a ray perpendicular to the xy plane is extended along the z-axis, and its points of intersection with the surface polygons of the object are computed, as depicted in Fig. 4. These intersection points are sorted along the direction of the ray. By connecting odd-numbered intersection points with even-numbered intersection points using lines, a set of z-axis-aligned dexel model equivalent to the polyhedral model is obtained. Other x-axis-aligned dexel model and y-axis-aligned dexel model are obtained by repeating the above process using square meshes on the yz plane or the zx plane.

Fig. 5(a) illustrates a Z-map based milling simulation result using a workpiece model and a cutter path. Z-map model of the workpiece is defined based on a square mesh given in the horizontal reference plane. The square mesh is aligned with respect to the x- and y-axis of the world coordinate frame. In the milling simulation using the transformed cutter path and vertical cutter, the z coordinate of the top side points of the vertical Z-map segments are modified and a milling result shape was obtained. The top surface of a Z-map model can be recognized as a group of points arranged according to a two-dimensional (2D) grid structure. Z-map model thus can be converted into a polygon mesh by appropriately connecting the adjacent points. Since the polygon mesh obtained in this method is an open shape, the ray algorithm cannot be applied as it is. To realize the conversion, we developed a method for obtaining a closed polygon mesh based on a Z-map model.

When the square mesh for defining the Z-map is given sufficiently wide on the reference plane, there exist grid points without height information (white points in Fig. 5(a)) around the grid points with proper height information (black points) representing the machining result shape. By giving a sufficiently small height value to the white grid points, it is possible to define three-dimensional (3D) points for all grid points and define a polygon mesh that completely covers the workpiece shape by properly connecting the points. A closed polyhedron is finally obtained by attaching a rectangle of the same size as that of the mesh in the bottom side of the Z-map, as depicted in Fig. 5(b). An inverse transformation $\mathbf{T}_i^{-1}$ is applied to the obtained closed polyhedral model, and a triple-dexel model representing the machined shape is computed by applying the ray algorithm to the model.

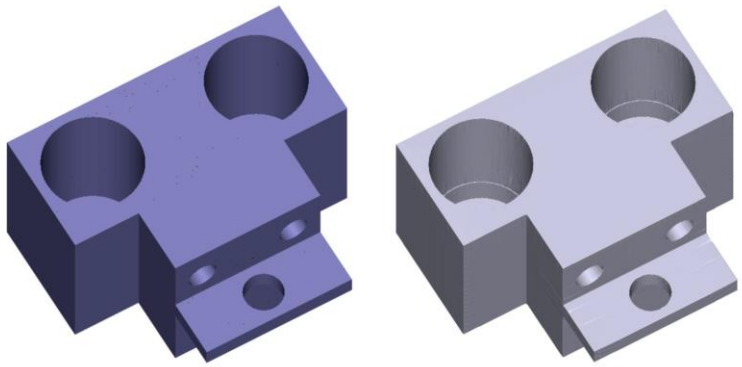

Fig. 6: (a) A machine part CAD model. (b) 3+2 axis machining simulation result using our software.

Computational Experiments

We implemented a software for visualizing 3+2 axis machining result using VisualStudio 2017. A 64-bit PC with an Intel Core i7 Processor, 16 GB memory, and an nVIDIA GeForce RTX-2080 GPU is used in the experiments. Fig. 6(a) illustrates a machine part CAD model of 30.0 x 26.0 x 17.0 mm size. A block model of a similar size (33.0 x 29.0 x 20.0 mm size) is used as the initial workpiece model. In this experiment, geometric simulations of 3+2 axis machining from 5 spindle directions (+z, +x, -x, -y and +y directions) are executed. For each 3+2 machining operation, five types of cutters (two flat end cutters of ϕ12mm and ϕ6mm, and three ball end cutters of ϕ6mm ϕ2mm and ϕ0.4mm) with proper holders are used.

In the Z-map based milling simulation, 4,000 x 4,000 resolution square mesh is used for defining the Z-map (resolution varies depending on the spindle direction). After the Z-map based simulation, obtained polyhedral Z-map model is transformed back to the original position and converted to a triple-dexel model. Triple-dexel model defined based on 959 x 661 x 1,092 resolution spatial grid is used in the computation. Boolean intersection operations are then executed with the initial workpiece triple-dexel model and 5 triple-dexel models corresponding to the 5 spindle directions. The result triple-dexel model is further converted to a polyhedron by using Marching-Cubes algorithm. Fig. 6(b) shows the result workpiece model after the 3+2 axis machining. Machining result shape is correctly visualized. Fig 7 shows two Z-map models representing the workpiece shape after 3+2 axis machining from +z direction and +y direction. Total 83.67s is necessary for the Z-map based milling simulation, conversion from the polyhedral Z-maps to triple-dexel models, and Boolean intersection computation of 6 triple-dexel models.

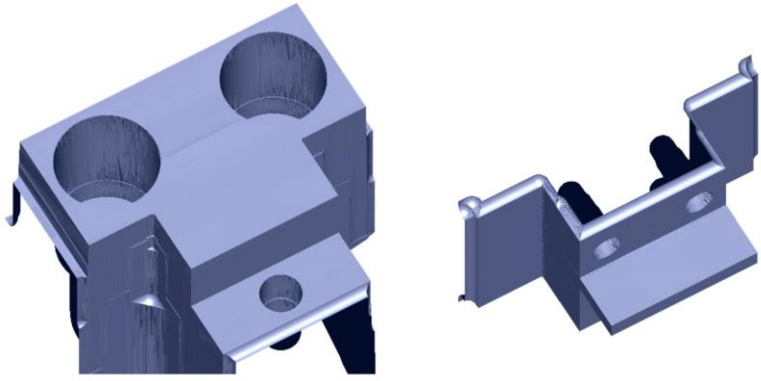

Fig. 7: Z-map based milling simulation results. (a) Machining result from +z direction. (b) Machining result from +y direction.

Conclusion:

In this paper, we propose a novel algorithm to realize geometric milling simulation of 3+2 axis machining. Using this algorithm, fast and accurate simulation is achieved by combining the Z-map based milling simulation technology and triple-dexel modeling. For each spindle direction, a Z-map based milling simulation is performed. The obtained machined shape is converted to a closed polyhedral model. Subsequently, it is further converted into a triple-dexel model using the ray algorithm. A resultant workpiece model reflecting all 3+2 machining operations is finally obtained by combining multiple machined shapes using the Boolean intersection operation. The effectiveness of the method is verified by computational experiments. Instead of using the depth buffer mechanism in the Z-map based milling simulation, we are planning to use the ray-tracing (RT) core of GPU in the computation, so that the processing speed can be further accelerated. The utilization of the RT core is also effective for the conversion of the polyhedral model to the triple-dexel model [3].

References:

[1] Benouamer, M.O.; Michelucci, D.: Bridging the gap between CSG and brep via a triple ray representation, Proceedings of ACM Symposium on Solid Modeling and Applications, 1997, 68–79. https://doi.org/10.1145/267734.267755

[2] Inui, M.; Ohta, A.: Using a GPU to Accelerate Die and Mold Fabrication, IEEE Computer Graphics and Applications, January/February 2007, 82–88. https://doi.org/10.1109/MCG.2007.23

[3] Inui, M.; Kaba, K.; Umezu, N.: Fast Dexelization of Polyhedral Models Using Ray-Tracing Cores of GPU, Computer-Aided Design and Applications, 18(4), 2021, 786-798. https://doi.org/10.14733/cadaps.2021.786-798.