Fast Simulation of Sculptured Surface Milling with 3-Axis NC Machine

Masatomo Inui (Masatomo.Inui@dse.ibaraki.ac.jp), Mitsuhiro Kaneda and R. Kakio
Department of Systems Engineering, Ibaraki University, Japan

Keywords: 3-axis milling simulation, dexel representation, box check

Abstract: Milling simulations are used prior to actual machining to detect potential problems of a tool path, for example excess milling of a workpiece. A simulation method based on the dexel representation of the workpiece is widely used. In this method, a milling process is geometrically evaluated by subtracting successive tool swept volumes along the path from a collection of thin vertical dexel cubes representing the workpiece shape. Two technologies for accelerating the dexel based milling simulation are proposed. (1) Most computation time in the simulation is elapsed in detecting such dexels intersecting the tool swept volumes. Based on some geometric characteristics of the tool path for milling the metal product, we define a tight bounding box to each surface element of the volumes which efficiently limits the range of potential intersecting dexels. (2) A critical milling process must be checked repeatedly from various viewing directions. The simulation method is further improved by integrating a history mechanism which records the computation results obtained in the first simulation process. By using the record, the computation time of repeated simulations can be substantially reduced.

1 Introduction

Molds and dies with free-form sculptured surfaces are usually fabricated using numerically controlled (NC) 3-axis milling machines with spherical cutters. A 3-axis NC machine drives a cutter according to a prescribed tool path, which is a set of continuous short linear trajectories. A tool path for milling a mold often contains more than hundred thousands trajectories. Most commercial CAM systems compute the tool path using the geometric model of designed surfaces in a fully or semi-automatic way. Numerical data describing the surface geometry are basically approximations and they often contain some inherent inconsistencies, for example small gaps between neighboring surfaces. Potential mistakes of inputting improper machining parameters to the systems are unavoidable. Therefore, generated tool paths could cause undesirable results of milling too deeply or leaving too much material.

1.1 Prior studies

In order to detect these problems, simulations and verifications of NC milling processes are needed prior to actual machining. A milling operation is geometrically equivalent to a series of Boolean subtractions of swept volumes of a cutter following a path from a solid model representing the stock shape. Based on this concept, many milling simulation systems have been developed. One of the major differences between them is the shape representation scheme of the tool swept volume and the workpiece.

In the early study, CSG modelers are directly used to generate shaded images of NC milling processes [5]. Atherton *et al.* [1], Van Hook [6] and Wang *et al.* [10] all use a projection of the workpiece shape in a fixed viewing direction and a variation of Z-buffer for visualizing the milling process. Different from the conventional Z-buffer which records only the Z height of the front visible surface for each ray through the pixels, Van Hook's dexel structure records the Z heights of the front and back surfaces

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35392-0_40

G. J. Olling et al. (eds.), *Machining Impossible Shapes*

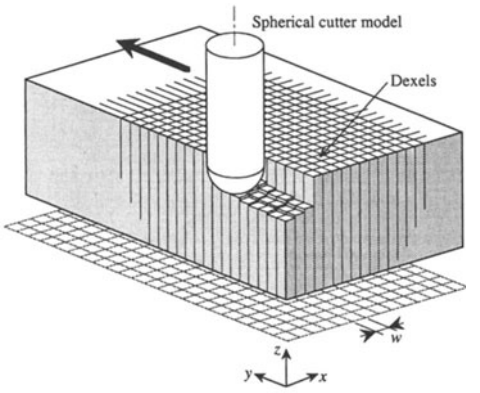

Figure 1: Illustration of the dexel based machining simulation of 3-axis milling.

of the objects. In this representation, Boolean subtraction is reduced to simple one-dimensional Z height comparisons.

Huang and Oliver [7] used a dexel representation derived from the dexel structure of Van Hook. A similar data structure is used by Saito and Takahashi [8] in their G-buffer method. A major distinction between their approach and the original dexel structure is that the ray direction for measuring the Z height is not limited to the viewing direction. In the 3-axis milling simulation, the spindle axis of the cutter is usually selected as the ray direction. This modification enables dynamic viewing transformations in the milling simulation.

1.2 Research purpose

Milling simulation method based on the view independent dexel representation is most commonly used in practice because of its computation speed. Figure 1 illustrates a simulation process using a workpiece model in the dexel representation. As shown in the figure, the workpiece shape is approximated as a collection of thin vertical dexel cubes. For each tool movement, intersections between the dexels and the tool swept volume are computed and removed from the workpiece model. The dexel representation is not suitable for the visualization because of its inevitable staircase errors on the workpiece surface. In order to realize a natural shaded display, small triangular patches are added to smoothly wrap the surface.

The dexel based simulation is faster than the other methods, however it is still difficult to complete the whole simulation of a complicated milling process in an interactive time period. In order to further accelerate the simulation, the following two technologies are proposed in this paper.

Improved bounding box: Most computation time in the simulation is elapsed in detecting such dexels intersecting the tool swept volumes. CAM systems usually generate a tool path so that its component linear trajectories satisfy some machinability conditions of the metal product. Based on the conditions, our simulation method defines a tight bounding box to each surface element of the

tool swept volumes, which efficiently limits the range of potential intersecting dexels.

History mechanism: In order to detect milling troubles such as gouges, critical milling processes must be checked repeatedly from various viewing directions. Our simulation method is improved by integrating a history mechanism which records the computation results obtained in the first simulation process. By using the record, the computation time of repeated simulations can be substantially reduced.

Based on these technologies, an experimental milling simulation system is implemented and demonstrated.

The organization of the paper is as follows. Some details of the dexel based milling simulation are given in the second section. Definitions of the improved bounding box and the history mechanism are explained in the third and fourth sections respectively. Computation results of the experimental system are given in the fifth section.

2 Dexel Based Milling Simulation

Consider 3-axis NC milling with a spherical cutter of radius r. As the reference of the workpiece shape and the cutter position, the world coordinate frame is placed so that its z-axis becomes parallel to the spindle axis of the cutter (see figure 1). In the following discussion, we assume that the workpiece has monotone shape in the z-axis direction. In the other words, the intersection between an arbitrary line in the z-axis direction and the workpiece is either a line segment, a point, or empty. Stock objects for molds and dies usually have rectangular shape and they satisfy this condition. In our coordinate frame specification, milling operations using spherical cutters remove the workpiece material from the z-axis direction. The workpiece thus retains the characteristic of being z-monotone after any 3-axis milling operations are applied to it.

2.1 Workpiece representation

Dexels are computed based on a two-dimensional regular square grid in the xy-plane as shown in figure 1. The grid is aligned with respect to the x- and y-axis of the coordinate frame. It is positioned so that its boundary encloses the projection of the stock object to the xy-plane. For each grid point, a ray extending along the z-axis direction is considered. The intersection segment between the ray and the stock object is computed and it is replaced to a dexel cube of the same length. Since the stock object is z-monotone, at most one dexel is specified to each grid point.

The accuracy of the simulation is determined by the grid size w. Smaller size enables more accurate simulations, but it requires an excessive amount of the storage. Therefore, the grid size is determined by a compromise between the accuracy and the practical requirement of the simulation. In the finish milling of molds and dies, a spherical cutter of radius $1.0mm$ is typically used. We thus select $w = 0.2mm$ based on the work [4]. This grid resolution gives a good result in the simulation of milling rather flat surfaces, however it is not sufficiently fine for evaluating the machined surfaces which happen to be parallel to the rays [7].

A grid point with the minimum x and y coordinates is selected as the reference point. Each grid point is addressed by an integer pair $(i,\ j)$ representing the number of grid intervals counting from the reference point (0, 0) in the x- and y-axis directions respectively. Coordinates of $(i,\ j)$ grid point in the xy-plane become $(x_{ij},\ y_{ij}) = (x_{00} + w \times i,\ y_{00} + w \times j)$ where $(x_{00},\ y_{00})$ means the coordinates of the reference point. A dexel corresponding to grid point $(i,\ j)$ is referred to $\mathbf{d}_{ij}$ in the following discussion.

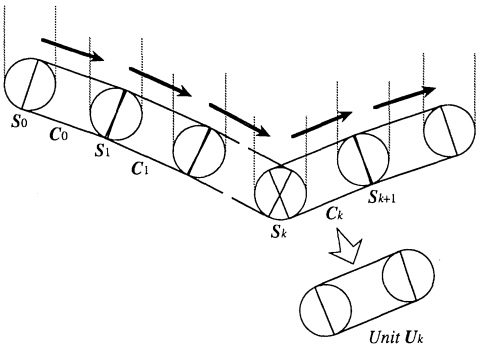

Figure 2: Series of swept volumes of a spherical cutter following a tool path and a unit volume corresponding to a single linear movement.

2.2 Milling simulation using units

A 3-axis milling process can be simulated by iteratively subtracting a tool swept volume corresponding to each linear trajectory of a tool path from the dexels. Since the workpiece shape is z-monotone, the subtraction is equivalent to an update of the top height of each dexel if the tool swept volume passes over it and the bottom surface of the volume cuts the dexel lower than the current height.

The bottom of the swept volume is generated by the spherical portion of a cutting tool. We thus consider the dexel modification caused by a swept volume of a sphere whose radius is the same as that of the tool. Figure 2 illustrates volumes obtained by sweeping the sphere along a tool path. Since the tool path is a collection of continuous line segments, the swept volumes become a series of cylinders smoothly jointed with spheres of the same radius as shown in the figure. We number them according to their appearance order in the swept volumes. In the following discussion, $\mathcal{S}_k$ and $\mathcal{C}_k$ mean the k-th sphere and k-th cylinder in the whole tool swept volumes respectively. For each linear movement of the tool, we can consider a straight swept volume of a sphere called "unit". The k-th unit $\mathcal{U}_k$ has a Boolean union shape of two spheres $\mathcal{S}_k$ and $\mathcal{S}_{k+1}$ and a cylinder $\mathcal{C}_k$ (see figure 2).

By using the terms defined above, we can describe a 3-axis milling simulation algorithm in a pseudo-code style as follows;

Step 1 Generate a dexel model of the initial stock object.

Step 2 Iterate the following operations for each unit $\mathcal{U}_k$ along a tool path.

2.1 Iterate the following operations for each dexel $\mathbf{d}_{ij}$.

2.1.1 Check intersections between dexel $\mathbf{d}_{ij}$ and a unit $\mathcal{U}_k$.

2.1.2 If they have an intersection, cut the intersection portion of $\mathbf{d}_{ij}$ by updating its height (see figure 3).

2.3 Box check

Detection of dexels intersecting each unit is the most critical step to realize the fast milling simulation. A unit has a cylindrical pin shape bounded by two half spheres

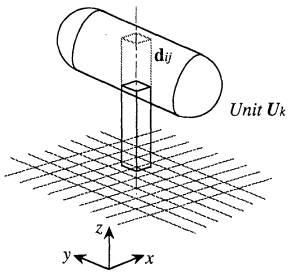

Figure 3: Cutting a dexel $\mathbf{d}_{ij}$ with a unit $\mathcal{U}_k$.

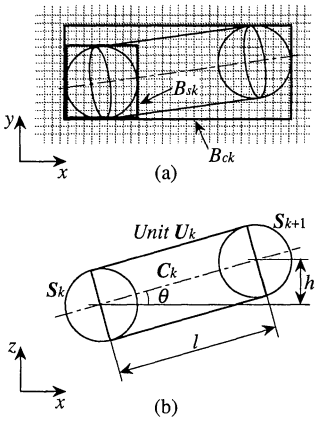

Figure 4: Simple bounding boxes B_{sk} and B_{ck} defined to a half sphere $\mathcal{S}_k$ and a cylindrical surface portion $\mathcal{C}_k$ of a unit $\mathcal{U}_k$ respectively.

at its ends as shown in figure 3. Therefore, the detection task can be decomposed into two sub tasks, which are (1) detection of dexels intersecting the half sphere surfaces of a unit, and (2) detection of dexels intersecting the cylindrical surface portion of it. Dexels are generated based on a regular square grid in the xy-plane. This definition enables to design a smart detection algorithm which only checks such dexels whose corresponding grid points lie under the projections of the half spheres and the cylindrical surface portion to the xy-plane.

Potential intersecting dexels are thus efficiently localized by using a two-dimensional box bounding the projection of each surface element of a unit [4]. The following is a typical definition of bounding boxes;

For half sphere: An axis-aligned square holding the projection of the complete sphere of radius r within.

For cylinder: An axis-aligned rectangle holding the projection of the entire unit within.

In figure 4(a), some bounding boxes defined to surface elements of a unit $\mathcal{U}_k$ are illustrated. A square B_{sk} represents the bounding box for a half sphere corresponding to $\mathcal{S}_k$, and a rectangle B_{ck} represents the bounding box for a cylinder $\mathcal{C}_k$ given in figure 4(b).

3 Improvements of Bounding Boxes

The accurate and high quality surface generation is the most critical requirement in milling metal products especially molds and dies. In order to satisfy the requirement, some machinability conditions of the metal are considered in the tool path generation. Based on the conditions, definitions of the bounding boxes are improved so that they can localize the potential intersecting dexels more tightly.

3.1 Geometric characteristics of tool paths

The following two conditions are usually considered in generating tool paths for milling metal products [3];

Smooth cutting load: Upward and downward motions of a cutter cause cutting depth fluctuations and load irregularity in milling, and they consequently hinder the accurate surface generation. Therefore, the cutter should be controlled to move horizontally as much as possible.

Smooth tool path: Sharp turning of a cutter in milling pushes the cutter off the course and it would leave tool marks on the machined surface. In order to avoid these errors, a tool path is generated so that successive component segments of the path are connected smoothly.

Because of the smooth cutting load condition, popular zig-zag milling is suitable only for machining rather flat surface portions of molds and dies. The contour milling method generally gives better results in machining other wall shapes of form features. In these milling methods, a cutter traces a (near) horizontal tool path. Most units generated by a sphere moving along such a path are thus horizontally positioned like a unit of $\theta \approx 0$ in figure 4(b).

Bounding boxes given in the previous section are inappropriate for localizing the dexels intersecting such a horizontal unit. The projection of a half sphere portion of a horizontal unit to the xy-plane becomes a half moon shape. Dexels whose corresponding grid points are contained within this projection can intersect the half sphere surface. Since the square box enclosing the complete circle of radius r is nearly two times larger than the half moon shape, many non-intersecting dexels are selected as the potential intersecting ones.

Simple bounding boxes for the cylindrical surface portion of horizontal units cause similar problems in the dexel localization. The projection of a horizontal cylinder has a rectangular shape. A bounding box defined based on the projection of the entire unit generally has much larger area than this shape. Relative area difference between the simple box and the projected cylinder becomes even larger for short units, which typically appear in the milling simulation of sculptured surfaces.

3.2 Improved bounding boxes

We solve these problems by properly modifying the bounding box definitions so that the box can hold the projection of each surface element of a unit more tightly. Some additional computations are needed for generating bounding boxes in the new

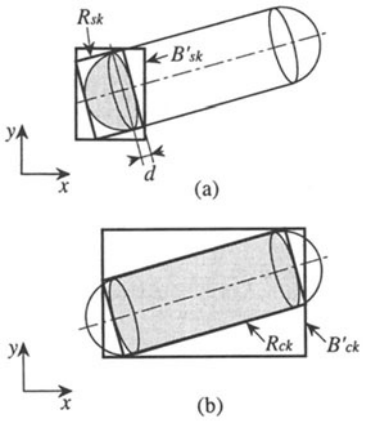

Figure 5: Definitions of boxes for the projection of a half sphere (a) and for the projection of a cylindrical surface portion of a unit (b).

definitions, however total computation time of the milling simulation can be reduced consequently because they can save time-consuming dexel-unit intersection detections.

Gray zone in figure 5(a) represents the projection of a half sphere portion of a unit. In order to efficiently localize the grid points being contained within this zone, the smallest rectangle which holds the projection within is defined. See rectangle R_{sk} in the same figure. An axis-aligned box which tightly encloses the rectangle defined above is then derived. B'_{sk} in figure 5(a) corresponds to this box. Boolean intersection of the axis-aligned box B'_{sk} and our prior bounding box with square shape (B_{sk} in figure 4(a)) is used as the new two-dimensional bounding box of the half sphere portion. Rectangle R_{sk} can be computed easily if distance d in the figure is known. This distance is derived as $d = rh/l$ where l and h mean the distance between the center points of two end half spheres of the unit and their distance in the vertical direction, respectively (see figure 4(b)).

By using a similar method, two-dimensional bounding box for the cylindrical surface portion of a unit is improved. Gray zone in figure 5(b) represents the projection of the cylindrical surface portion. The smallest rectangle which holds the projection within is defined at first. R_{ck} in figure 5(b) represents this rectangle. An axis-aligned box which tightly encloses the rectangle defined above is then derived. See box B'_{ck} in the same figure. Boolean intersection of the axis-aligned box B'_{ck} and our prior bounding box enclosing the projection of the entire unit (B_{ck} in figure 4(a)) is used as the new bounding box of the cylindrical surface portion.

3.3 Milling simulation using chains

In sculptured surface milling, a cutter iterates very short linear motions. Since each motion of the cutter removes very small amount of the workpiece material, execution of the dexel modification for every single cutter movement is not necessary. Therefore, we modify the milling simulation algorithm so that the dexel modification is executed for every $10 \sim 50$ tool movements. A swept volume of a sphere moving along a series of tool trajectories is called "chain" in the following discussion. A

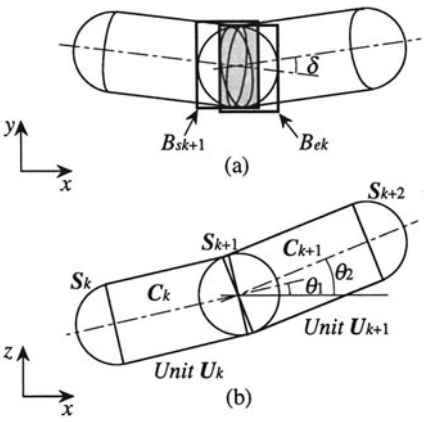

Figure 6: A bounding box defined to a spherical surface portion shared by two successive units $\mathcal{U}_k$ and $\mathcal{U}_{k+1}$.

chain has a Boolean union shape of a collection of units corresponding to the tool trajectories. New simulation algorithm is basically the same as the algorithm given in the previous section, but a chain of $10 \sim 50$ successive units is subtracted from the dexels in each dexel modification step.

3.4 Bounding box for a shared sphere

Milling simulations using chains can be performed more efficiently than simulations using units, because further localization of the dexels intersecting each surface element of a chain is possible. Let us explain this improvement using a simple chain with only two successive units $\mathcal{U}_k$ and $\mathcal{U}_{k+1}$ given in figure 6. Figure 6(a) illustrates a projection of the chain to the xy-plane, and (b) illustrates another projection of the same chain to the xz-plane. This chain has five surface elements, which are three spherical surfaces corresponding to $\mathcal{S}_k$, $\mathcal{S}_{k+1}$, and $\mathcal{S}_{k+2}$ and two cylindrical surfaces $\mathcal{C}_k$ and $\mathcal{C}_{k+1}$.

As shown in figure 6, spherical surface portion appearing between two successive cylinders $\mathcal{C}_k$ and $\mathcal{C}_{k+1}$ has very small area, therefore its intersecting dexels are very limited. We can properly localize these dexels by using a two-dimensional bounding box defined in the following manner. This spherical portion can be considered as a part of the half sphere of unit $\mathcal{U}_k$. It can be considered as a part of the half sphere of the next unit $\mathcal{U}_{k+1}$ also. Therefore, grid points of dexels intersecting this shared spherical surface must be contained within a box $B_{e\,k}$ defined for the half sphere of $\mathcal{U}_k$ and another box $B_{s\,k+1}$ defined for the half sphere of $\mathcal{U}_{k+1}$. The intersection of these two boxes (gray zone in figure 6(a)) is thus used as the two-dimensional bounding box for the shared spherical surface.

Because of the smooth cutting load condition, most component units of a chain are horizontally positioned like units of $\theta_1 \approx 0$ and $\theta_2 \approx 0$ in figure 6(b). Because of the smooth tool path condition, the angle between two successive units (angle δ in the figure) becomes small also. In these geometric conditions, the bounding box for the shared spherical surface becomes very small and it can localize the potential

intersecting dexels very tightly.

4 History Mechanism

Molds and dies often contain many complicated concave features. Milling of these features is prone to gouging troubles, therefore such milling process must be checked repeatedly from different viewing directions. In order to realize the efficient multiple check of the same milling operation, simulation systems must provide "back" function and "redo" function of the simulation process. The back function recovers the workpiece shape in a certain prior state, and the redo function repeats the same milling simulation from the recovered state.

Milling simulation algorithms given in the previous sections modify the workpiece model in a destructive manner, therefore re-execution of the whole simulation process from the initial stock shape is necessary to recover a workpiece state. It needs the computation time proportional to the number of total tool movements in the worst case. In order to realize the fast recovery of any prior workpiece states, a history mechanism is integrated in our simulation system. This mechanism records the modification history of the workpiece model during the first simulation process. By using the history data, any workpiece state in the milling process can be recovered with some limited computations.

4.1 History recording method

In the following discussion, milling simulations using chains are assumed. A workpiece modification caused by tool movements corresponding to a single chain is usually localized and limited number of dexels are modified in each simulation step. Therefore, the average number of modifications of each dexel occurred in the whole simulation process becomes small compare to the total number of the simulation steps. Based on this consideration, our mechanism maintains the workpiece modification history by recording the shape (= top height) change history of each dexel.

For each dexel $\mathbf{d}_{ij}$, a stack s_{ij} for recording its history is prepared. The milling simulation algorithm is modified as follows so that the history stack of each dexel is updated when the top height of the dexel is changed.

Step 1 Generate a dexel model of the initial stock object.

Step 2 Initialize the history stacks of all dexels.

Step 3 Iterate the following operations for each chain $\mathcal{CH}_k$ along a tool path.

- **3.1** Select potential intersecting dexels using bounding boxes defined to surface elements of $\mathcal{CH}_k$.
- **3.2** Iterate the following operations for each selected dexel $\mathbf{d}_{ij}$.
  - **3.2.1** Check intersections between dexel $\mathbf{d}_{ij}$ and a chain $\mathcal{CH}_k$.
  - **3.2.2** If they have an intersection, cut the intersection portion of $\mathbf{d}_{ij}$ by updating its height. A pair of the updated height of $\mathbf{d}_{ij}$ and index k of chain $\mathcal{CH}_k$ is then pushed to the history stack s_{ij}.

Consider a case that the height of a dexel $\mathbf{d}_{ij}$ is modified to z_{k0} by a chain $\mathcal{CH}_{k0}$. The height is further modified to z_{k1} and z_{k2} by chains $\mathcal{CH}_{k1}$ and $\mathcal{CH}_{k2}$ successively. After the simulation, a history information $(k2,\ z_{k2}) \rightarrow (k1,\ z_{k1}) \rightarrow (k0,\ z_{k0})$ is stored in the history stack s_{ij} of dexel $\mathbf{d}_{ij}$.

4.2 Workpiece recovering method

Let us explain the history based back function by using a case of recovering the workpiece shape just after the dexel modification caused by a chain $\mathcal{CH}_k$. This recovery is achieved by checking the history stack of each dexel of the workpiece model in a downward manner from the top item of the stack. For each item in the stack, stored index value and index k of chain $\mathcal{CH}_k$ are compared. This comparison is iterated until an item whose index value is less than or equal to k is found, which must be the item in association with the largest index satisfying the inequality condition in the stack. Top height of the dexel is updated to the height value stored in the detected item. In our history stack example $(k2,\ z_{k2}) \rightarrow (k1,\ z_{k1}) \rightarrow (k0,\ z_{k0})$, the height of the dexel $\mathbf{d}_{ij}$ is updated to z_{k1} if k satisfies inequalities $k2 > k \geq k1 > k0$.

5 Computational Experiments

An experimental milling simulation system with proposed technologies is implemented using C language, and some computational experiments are performed. All NC codes and workpiece data used in the experiments are real data generated for rough milling, semi-finish milling, and finish milling of some mold parts. Dexel models of stock objects are generated based on a regular square grid of interval $0.2mm$. Each dexel modification in the simulation is executed using a chain with 40 successive units. All experiments are performed using SGI Onyx2 workstation (CPU: R10000 195MHz, Memory 512MB).

Table 1: Computation results.

Number of linear trajectories	Tool radius (mm)	Simple box (sec.)	New box (sec.)	With record (sec.)	Average stack depth
9,542	5.0	11.69	5.32	8.34	5.05
35,391	3.0	13.86	5.20	9.65	5.78
126,719	1.0	4.08	2.48	6.95	1.07
156,799	1.0	6.15	3.77	10.61	2.59

Some results of the experiments are described in table 1. The first column gives the number of linear trajectories in the path, and the second column gives the radius of the spherical cutter used in the simulation. The elapsed simulation time using the simple box check is given in the third column. It can be reduced to the time given in the fourth column by using our improved box check. In these computations, the history of the workpiece modification is not recorded. As shown in the table, the improved box check accelerates the simulation program two times faster than the program using the simple box check. A complex milling simulation with 150,000 tool movements can be executed in less than 5 seconds. Figure 7 illustrates a tessellated workpiece model after applying a milling simulation corresponding to the fourth row of table 1.

In the fifth column of table 1, the elapsed time for recording the history of the workpiece modification is given. The average depth of the history stacks is described in the sixth column. These results show that small additional time is enough for recording the whole history of the milling process. Required storage for the history is several times more than the storage needed in the usual simulation. Once the history is recorded, any workpiece shape in the milling process can be recovered instantly.

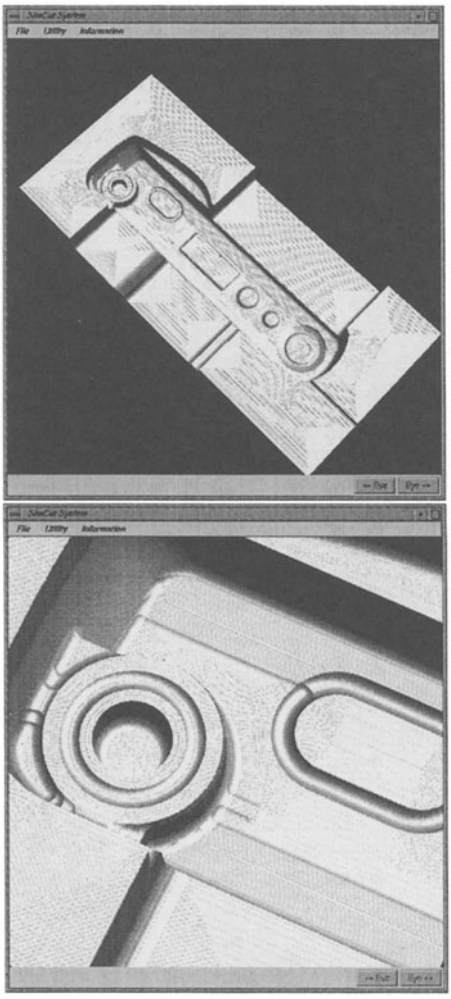

Figure 7: Simulation result of milling a mold part and its close-up picture.

6 Conclusions

Two technologies for accelerating the dexel based milling simulation are proposed, which are;

- Improved bounding boxes based on the geometric characteristics of the tool path for milling the metal product.
- Stack based history mechanism for reducing the computation time of repeated simulations of the same milling process.

Based on the technologies, an experimental milling simulation system is implemented and demonstrated.

The following enhancements of the algorithm are considered as our future research subjects.

- Tessellation algorithms of a solid object in a discrete representation are well studied in the computer graphics field, but most of them are not suitable for visualizing the milling result. Fast algorithms for naturally tessellating the dexel model are needed.
- Geometric simulation is not sufficient for validating the tool path data. Other physical properties such as cutting load and tool deformations must be checked in advance to realize high speed and accurate milling. Development of fast and reliable algorithms for analyzing these physical properties is our research subject also [9].

Acknowledgment

This research work is financially supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan (Grant No. 08650307). We thank Mr. Hideki Tamura of Ricoh Co., Ltd. and Mr. Yuji Hara of Graphic Products Co., Ltd. for their providing us the NC milling data.

References

[1] Atherton, P., Earl, C. and Fred, C.: A Graphical Simulation System for Dynamic Five-Axis NC Verification, *Proc. Autofact '87*, 1987, 2-1–2-12.

[2] Benouamer, M. O. and Michelucci, D.: Bridging the Gap between CSG and Brep via a Triple Ray Representation, *Proc. of Fourth Symposium on Solid Modeling and Applications*, 1997, 68–79.

[3] Choi, B. K., Kim, D. H. and Jerard, R. B.: C-space Approach to Tool-Path Generation for Die and Mould Machining, *Computer-Aided Design*, 1997, 29(9), 657–669.

[4] Drysdale, R. L., Jerard, R. B., Schaudt, B. and Hauck, K.: Discrete Simulation of NC Machining, *Algorithmica*, 1989, 4(1), 33–60.

[5] Fridshal, R., Cheng, K. P., Duncan, D. and Zucker, W.: Numerical Control Part Program Verification System, *Proc. Conf. on CAD/CAM Technology in Mechanical Engineering*, 1982, 236–254.

[6] Van Hook, T.: Real-Time Shaded NC Milling Display, *Computer Graphics*, 1986, 20(4), 15–20.

[7] Huang, Y. and Oliver, J. H.: NC Milling Error Assessment and Tool Path Correction, *Computer Graphics Proceedings*, 1994, 287–294.

[8] Saito, T. and Takahashi, T.: NC Machining with G-Buffer Method, *Computer Graphics*, 1991, 25(4), 207–216.

[9] Takata, S., Tsai, M. D., Inui, M. and Sata, T.: A Cutting Simulation System for Machinability Evaluation Using a Workpiece Model, *Annals of the CIRP*, 1989, 38(1), 417–420.

[10] Wang, W. P. and Wang, K. K.: Geometric Modeling for Swept Volume of Moving Solids, *IEEE Computer Graphics and Applications*, 1986, 6(12), 8–17.