Determination of Workpiece Postures for Three-Axis Machining Using Milling Simulation Results

Masatomo Inui[1], Sunao Kawai[2] and Nobuyuki Umezu[3]

[1]Ibaraki University, masatomo.inui.az@vc.ibaraki.ac.jp
[2]Ibaraki University, j94820c71@gmail.com
[3]Ibaraki University, nobuyuki.umezu.cs@vc.ibaraki.ac.jp

* Corresponding author: Masatomo Inui, masatomo.inui.az@vc.ibaraki.ac.jp

Abstract. This study describes a software system that determines the minimum combination of workpiece postures necessary for cutting a machine part. In a previous study, the authors developed a system for detecting difficult-to-machine shapes by repeating cutting simulations while changing the posture of the workpiece model. The newly developed technology extends this method to automatically determine the necessary workpiece postures by appropriately incorporating the simulation results. Six workpiece postures are commonly considered when machining mechanical parts. Geometric cutting simulations were performed for each posture, and for each polygon in the part model, thereby recording the possible workpiece postures for machining. Finally, the combinations of workpiece postures that allowed all polygons to be machined were examined, outputting the smallest combination as the calculation result. Computational experiments were conducted on several part models, confirming that appropriate workpiece posture combinations can be obtained.

Keywords: Manufacturability evaluation, geometric cutting simulation, computer-aided process planning.
DOI: https://doi.org/10.14733/cadaps.2025.107-118

1 INTRODUCTION

The increasing performance of Internet technology and the computerization of manufacturing processes have brought about various changes in the manufacturing industry. In a new business model that has been gaining popularity for the manufacturing of small quantities of mechanical parts, the machining of parts is ordered online and delivered quickly (e.g., MISUMI's Meviy System [15]). After a computer-aided design (CAD) model of the part is uploaded according to a specified procedure, an estimate of the time and price required for machining is displayed, and if it meets the requirements, the order is placed immediately. The parts are processed in the factory and sent by a courier service to the specified location for delivery at the specified date and time (Fig. 1). However, for such a business to be viable, the cost of machining the part must be estimated as quickly and accurately as possible based on a given CAD model.

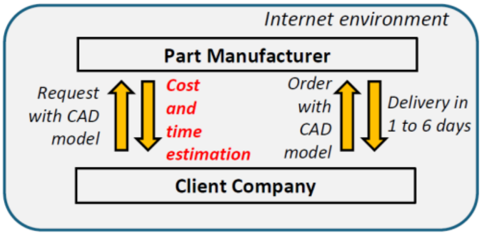

Figure 1: Online mechanical part manufacturing business.

In this type of part manufacturing, the available machining methods are usually limited to cutting, and the tools and workpiece mounting used for machining are also limited to standard methods, which are used to determine the machinability of the uploaded CAD model. The number of mounting changes in the workpiece, type of tooling used, and necessary machining time should also be determined in the shortest possible time to accurately estimate the costs. The ultimate goal of this research is to automate machining cost estimation, and changing the mounting of workpieces is a manual process that has a significant impact on machining costs. In this study, a new method was developed to automatically determine the workpiece mounting required to machine a part.

Our laboratory has already developed machinability evaluation software that determines whether a given CAD model is machinable using prepared cutting tools [9,10]. The software repeatedly simulates the cutting operations using these tools. If any detected shape cannot be removed by the cutters, the system determines that part as difficult to machine; otherwise, the part is recognized as machinable. In the small-lot production of mechanical parts, parts are usually machined from rectangular materials. Therefore, it is standard practice to place any of the six faces of a rectangular object or the flat faces obtained through machining in contact with the table, thereby fixing them and repeating the machining operations. The software we developed performs machining simulations while changing the six workpiece postures according to this standard mounting method. Using this software, difficult-to-machine shapes can be readily detected and visualized.

This software was extended to record the mounting position of each point on the surface of the part during machining simulations. For machinability evaluation, machining simulations were first performed with six standard mounting postures. After processing, the recording of the mounting postures at each point on the part surface was examined to determine the combination of mounting postures required to machine all the points on the surface of the part. Subsequently, the smallest combination of workpiece postures that can fully machine all points was determined and output as the result. The software was implemented and used to perform computational experiments on 20 parts to verify the effectiveness of the proposed method. The additional processing time required for posture determination was negligible.

In the next section, the authors briefly describe the previous studies in this field. Section 3 describes the machinability evaluation software based on the proposed cutting simulation technique. Section 4 describes the proposed method for determining the required combination of mounting postures based on the recorded cutting simulation data. Section 5 describes the results of the computational experiments and Section 6 presents the conclusions and future research targets.

2 PRIOR STUDIES

Determining the posture of a workpiece during machining has been a subject of research in process planning automation [1]. Conventional research usually employs the following methodologies. The basic idea is to extract characteristic partial shapes referred to as form features, such as holes, grooves, and pockets, in a part-shape model, thereby determining the spindle direction of the tool that can machine these shapes based on the geometric constraints of each form feature, finally deriving the workpiece posture based on the tool direction. Therefore, in previous studies, the main focus of research was on form feature extraction techniques and the determination of machinable tool directions from the extracted form features.

Numerous studies have been conducted on form feature extraction [20]. There are two major categories in this field: methods that detect form features as a set of faces or as volumetric objects. The former extracts features based on the connections between the faces of the part shape model [3,5,11,21]. As the spindle direction of the tool during machining is often determined from the geometric properties of the faces, this method has the advantage of facilitating the determination of the machining method after extraction. However, the same shape often has different types of surfaces and connections (e.g., whether the sides of a hole are represented by a single cylindrical surface or a group of long and narrow triangles), and it is difficult to detect form features when multiple form features are mutually connected (e.g., when a pocket shape has additional pockets at the bottom).

Another promising method for form feature extraction considers volumetric elements [18,13,17,19]. By considering the rectangular object that tightly contains the part shape as the initial material shape and performing a set operation to subtract the part shape from this rectangular object, the solid shape that should be removed by cutting (removal volume) can be obtained. By further dividing this solid model into basic volumes that can be machined by unit cutting operations such as drilling, slot milling, and pocket milling, key operations for process planning, such as essential machining operations and tool spindle direction in machining, can be automatically determined. For these reasons, the use of volumetric form features has increased in recent years in process planning research.

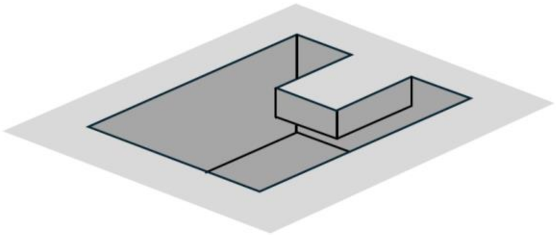

Figure 2: Pocket features with an unmachinable horizontal protrusion

Conventional systems based on form-feature extraction cannot correctly perform process-planning tasks when they find a shape in a part model that cannot be machined. For example, the pocket shape shown in Fig. 2 exhibits a horizontal protrusion on the side face. The lower part of this protrusion cannot be produced by cutting. Conventional methods cannot recognize this pocket shape with side protrusion and stop processing at this point, or they recognize the pocket shape and protrusion as separate features and erroneously think that the lower part of the protrusion can be machined from the side. In contrast, the proposed cutting-simulation-based method can correctly detect the lower part as a difficult-to-machine shape (concave corners consisting of mutually orthogonal faces are also difficult to machine for this pocket), thereby determining that the other shapes can be machined from the Z-axis plus direction. Therefore, the detection of difficult-to-machine shapes and workpiece posture determination can be performed simultaneously using the proposed method.

The problem of determining the mounting posture of a workpiece in three-axis machining is geometrically identical to that of determining the spindle direction of a tool in five-axis machining.

Most tool-direction-determination methods for the five-axis machining are specialized for impellers, molds with curved surfaces, and airplane parts, and do not target the machining of ordinary mechanical parts [2,4,12,14,16]. For example, in the five-axis machining, ball-end cutters are commonly used for machining curved surfaces, whereas flat- or radius-end cutters are often used in three-axis machining for the accurate and efficient production of horizontal and/or vertical surfaces of the mechanical part. In five-axis machining, the tool spindle direction can be freely changed; therefore, determining the optimal direction is important while considering geometric and mechanical constraints. In this research, the objective was to obtain the minimum combination of the required workpiece postures from six possible postures to reduce the manufacturing cost of the part. Thus, the objectives of this research are different from those of conventional five-axis machining, and solving this problem using the research results for five-axis machining is difficult.

3 MACHINABILITY EVALUATION USING CUTTING SIMULATIONS

The proposed method uses previously developed software to detect difficult-to-machine shapes [9, 10]. This software assumes that the three-axis cutting operation is used as the machining method, and the cutting tools are limited to predefined ball-, flat- and radius-end cutters. Information on the available cutters (cutting edge type, cutter shape, shank, and holder) is provided beforehand. The polyhedral model of the machine part is provided as input data. STL model was assumed in the current implementation, but any data format can be used as long as it is in the B-reps format, which allows polyhedralization. An axis-aligned bounding box (AABB) that encompasses the part shape is considered as the initial workpiece shape, and a local coordinate frame consisting of coordinate axes parallel to each edge of this rectangular object is fixed to the part.

Assuming that the cutting operation is performed by mounting a workpiece in such a way that any one of the +X, -X, +Y, -Y, +Z, or -Z axes of the model coordinate frame points upward, the machining simulation is repeatedly performed for each workpiece posture while changing the prepared tools. After the cutting simulations, the remaining shapes to be machined are examined, and the surfaces of the machine parts corresponding to these remaining shapes are painted in different colors to visualize the difficult-to-machine shapes. The processing flow of the algorithm is provided below (Fig. 3).

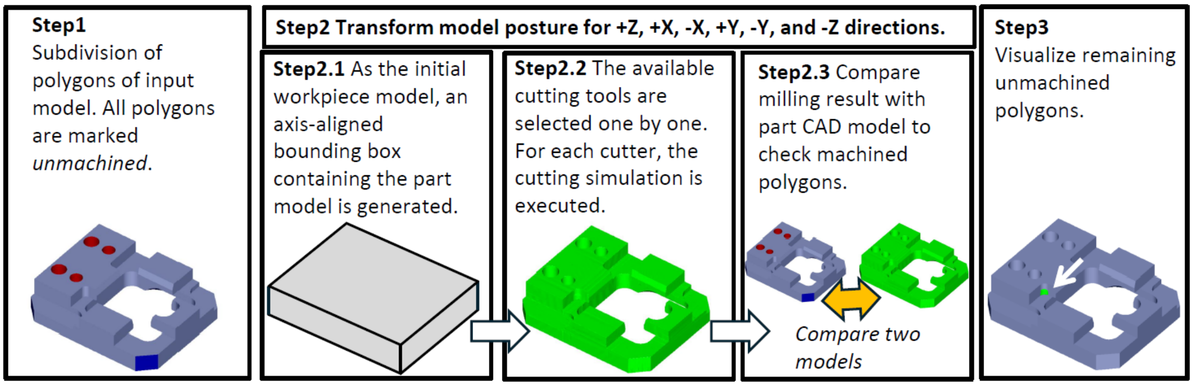

Figure 3: Simulation-based difficult-to-machine shape detection method.

Step 1: Polygons covering the part surface are subdivided such that sufficiently fine polygons of approximately the same size cover the surface of the part. All polygons are then marked as *unmachined*.

Step 2: The posture of the part model is transformed in such an order that either one of the +Z, +X, −X, +Y, -Y, and −Z axis directions in the local coordinate frame of the model points upward. Subsequently, the following steps are executed for each model posture:

Step 2.1: An AABB that contains the part model is generated. This box is considered the initial shape of the workpiece.

Step 2.2: The available cutting tools are selected individually, whereby for each cutter, the following operations are performed:

- The Minkowski sum of the part model and inverted shape of the selected cutter are computed. A zigzag-type cutter path is generated to machine the entire model based on the Minkowski sum shape.
- The geometric cutting simulation is performed using the obtained cutter path and tool shape data, modifying the shape model of the workpiece during the simulation.

Step 2.3: Following the simulations, the workpiece shape thus obtained is compared with that of the part model. If each polygon of the part model is machined accurately, the polygon mark is changed from *unmachined* to *machined*.

Step 3: The polygons with the remaining *unmachined* marks are painted green to highlight difficult-to-machine shapes on the model surface.

This software refers only to the shape of the part and does not consider the workpiece shape when generating tool paths. In actual machining, in certain cases the interference between the tool holder and the workpiece can complicate the machining; however, this software does not consider such cases, as explained in the concluding section.

The software determines difficult-to-machine shapes based on the polygon. Therefore, partitioning of the part surface in **Step 1** is performed so that the polygons after the subdivision process are sufficiently fine and their sizes are almost the same. Each polygon of the input model is examined, and when the length of the longest edge of the polygon is greater than the predetermined length *l*, the polygon is subdivided such that all edges of the polygon become shorter than *l*.

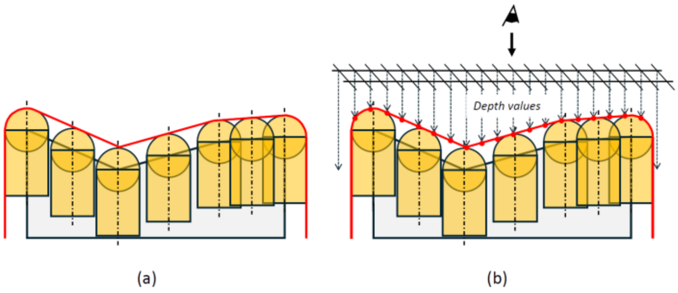

Figure 4: Calculation of the uppermost surface of the Minkowski sum shape of an inverted tool and part model using the depth buffer.

The tool path generation and cutting simulation in **Step 2.2** utilizes the GPU's depth buffer functionality. The tool path for three-axis numerical control (NC) machining can be generated on the uppermost surface of the Minkowski sum shape of an inverted tool and a part model. Such a surface can be computed using the GPU's depth buffer function by considering the swept volume of the inverted tool moving along the part’s surface (Fig. 4(a)), thereby generating a hidden-surface eliminated image of it looking down from above (Fig. 4(b)) [6]. Using the depth values obtained during image generation, the uppermost surface can be calculated. In actual NC machining, holes and chamfers are machined using drills and specialized tools. Flat-end cutters are only used for machining surfaces parallel or perpendicular to the direction of the spindle axis of the tool. Our system detects these characteristic features in advance and then machines them using a different approach. For details on these techniques, refer to our previous study [10].

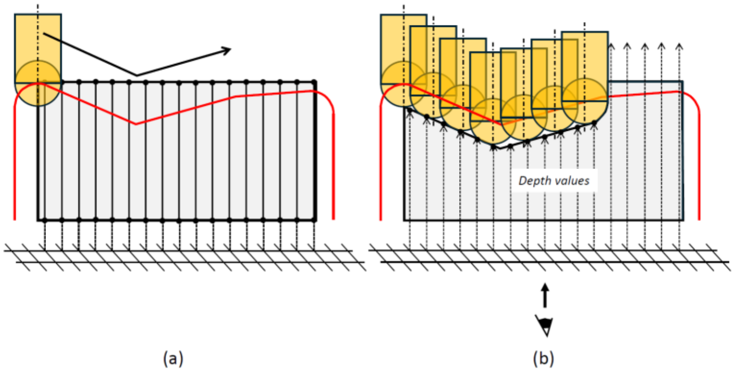

Figure 5: Calculation of the bottom surface of swept volume of a tool using depth buffer to realize geometric cutting simulation.

The geometric simulation results of NC machining can be obtained by considering the swept volume of the tool moving along the path and subtracting it from the three-dimensional (3D) model of the workpiece. Let us consider a projection of the workpiece geometry onto the XY plane and place a high-resolution Cartesian grid on the XY plane to encompass the projection. Considering the intersection of the workpiece shape with a half-line extending in the Z-axis direction from each grid point, the workpiece shape can be represented as a set of vertical line segments (Fig. 5(a)). Such a 3D model is known as a dexel model. In dexel modeling, the resulting shape of the three-axis NC machining can be computed by measuring the height of the bottom surface of the tool's swept volume for each grid point. This value can be obtained by generating a hidden-surface eliminated image of the tool's swept volume [7]. As shown in Fig. 5(b), the image is generated by looking up at the swept volume. Using these techniques, toolpath generation and machining simulations can be performed at high processing speeds.

After simulating the cutting using all the available tools, **Step 2.3** checks whether each polygon has been machined. If the material remaining on the polygon is sufficiently thin, it is judged to have been successfully machined, and the polygon's *unmachined* mark is changed to *machined*. Because the polygons are very small, the amount of remaining material is measured at the center of gravity of each polygon, such that if it is smaller than a specified amount s (0.02 mm in our current implementation), the polygon is judged to have been machined. The specific method is illustrated in Fig. 6. Point $\boldsymbol{p}$ is generated at a position displaced from the polygon's center of gravity $\boldsymbol{g}$ by s in the normal direction. The position of $\boldsymbol{p}$ is compared with the workpiece model dexels which represent the machining results. The coordinates of $\boldsymbol{p}$ are projected onto the XY plane, and the grid point closest to the projection of $\boldsymbol{p}$ is determined (Fig. 6(a)). Then, $\boldsymbol{p}$ is compared with the dexels extending from this point to determine whether the polygon has been machined. In Fig. 6(b), $\boldsymbol{p}$ is inside the dexel; thus, the polygon remains unmachined. In contrast, in Fig. 6(c), $\boldsymbol{p}$ is outside the dexel, and the polygon is judged to have been machined.

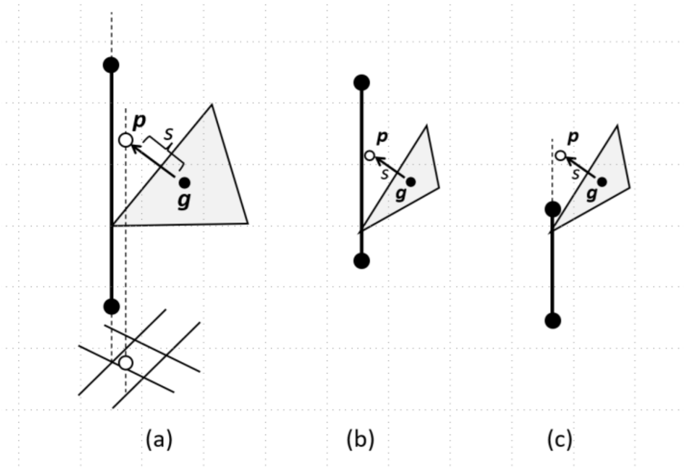

Figure 6: Detection method for machined polygons.

4 SELECTION OF NECESSARY WORKPIECE POSTURES

4.1 Recording of Machined/Unmachined Data

In this study, a simple dexel model was used to record the results of the geometric cutting simulation. Because dexel models represent 3D shapes as a group of line segments along the Z-axis, they cannot freely change their orientation. A dexel model representing the box-like workpiece shape is thus recreated whenever the model posture changes in the +Z, +X, −X, +Y, −Y, and −Z axis directions. After the cutting simulation, the part and machined workpiece models are compared to determine whether each polygon of the part model has been processed. Once the material is removed from the polygon, the *unmachined* mark of the polygon is changed to *machined*, and the cutting result is recorded for each polygon of the model.

Figure 7 illustrates this process. In preparation, all the surface polygons of the part model are marked *unmachined* (green color in the figure). In the initial machining, the model is oriented such that the +Z-axis points upward. A box-like workpiece shape is defined as the dexel model, and the cutter path computation and cutting simulations are performed. The machining result for each polygon is verified by comparing the part model with the machined workpiece model obtained by the simulation. The *unmachined* mark is changed to *machined* (gray) when material removal for the polygon is complete. The model is then set such that the +X-axis points upward. A box-like workpiece model is redefined for the new orientation, and the same process is repeated.

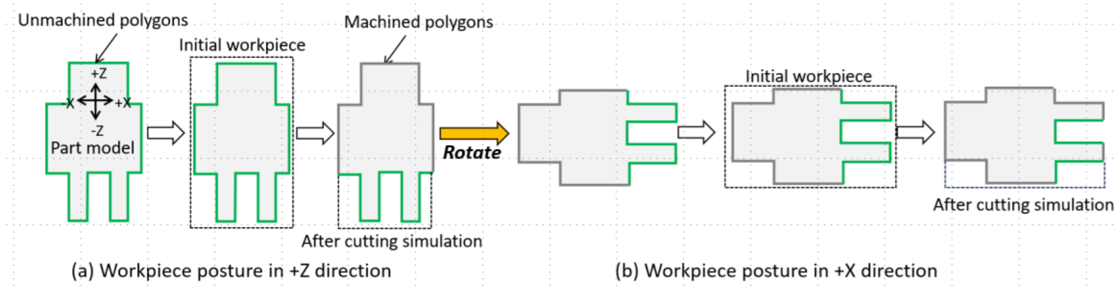

Figure 7: Machinable polygon detection process.

In this method, the workpiece model is recreated each time; therefore, previous machining results are not reflected in the workpiece model. However, each polygon of the part has a record of the *machined* or *unmachined* status; therefore, the polygons that remain *unmachined* until the end of the process correspond to difficult-to-machine shapes. A technique was developed to convert the dexel model into a polyhedron [8]. Using this technique, the orientation of a dexel model can be changed by converting the dexel model into a polyhedron and converting it back into a dexel model after changing its orientation. This technology enables the detection of difficult-to-machine shapes by considering the workpiece model in a work-in-progress state. It also enables machining simulations that consider collisions between tool holders and workpieces; however, this is not yet practical because of processing time issues.

4.2 Extension of Recording of Polygons

In the above software, the cutting simulations are repeated while changing the posture of the workpiece in six different directions. However, for many parts, machining can be completed with fewer postural changes. This is illustrated in Fig. 8 for a two-dimensional machining example. In this case, up to four posture changes are possible, with each of the +Z, +X, -X, and -Z axis pointing upward; however, in reality, machining can be completed with only two workpiece postures as illustrated in Fig. 8(b). In real machining, changing the workpiece posture is a time-consuming process that requires detaching and remounting the workpiece and subsequent adjustment of the origin. As these tasks are performed manually, they are prone to human error. Therefore, with fewer number of posture changes, faster, cheaper, and safer machining can be achieved.

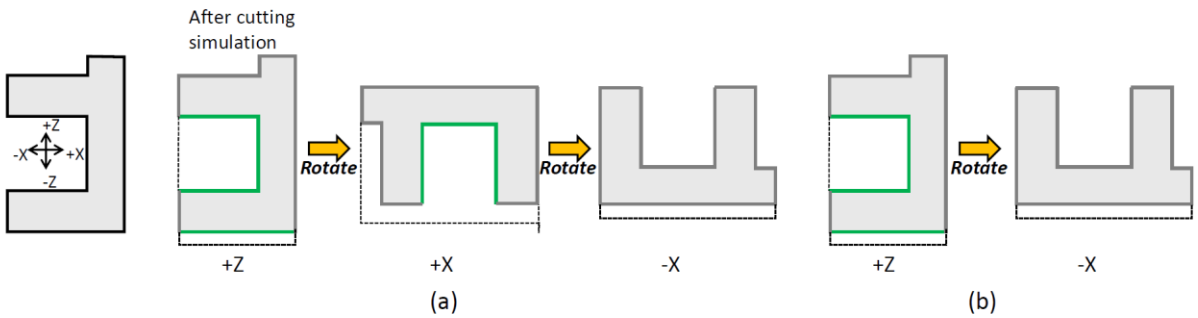

Figure 8: Machinable polygon detection with fewer posture changes.

The authors developed a technique to determine the minimum number of workpiece postures by extending our previous software to detect difficult-to-machine shapes. In conventional software, a cutting simulation is performed for each workpiece posture, whereby each surface polygon of the part model is checked and recorded to determine whether machining of the polygon is complete. In our new software, the recording method was modified to include workpiece posture information for each polygon that was determined to be machinable at a certain workpiece posture. In Fig. 9, a cutting simulation is performed for each workpiece posture in the +Z, +X, -X, and -Z axis directions, and the polygons for which machining is completed are recorded along with the workpiece posture information at that time. In this case, machining can be completed with only two workpiece postures, since the record of the workpiece posture for each polygon indicates that all polygons can be machined if the workpiece is fixed in the +Z-axis or -Z-axis direction (red text in Fig. 9). Fig. 10 confirms that machining of the part can be completed with the two workpiece postures.

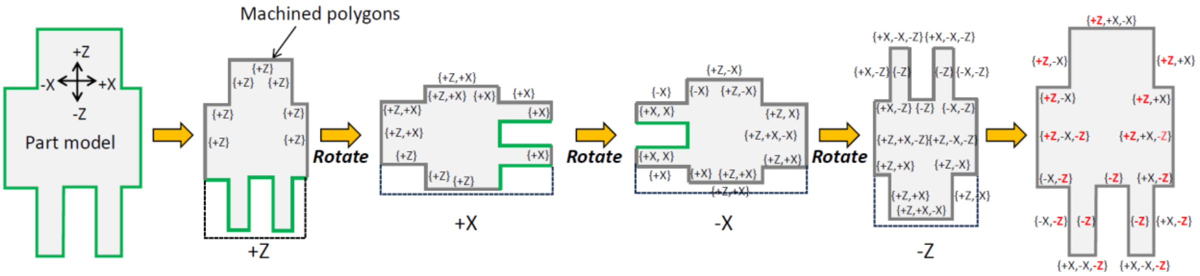

Figure 9: Recording of workpiece posture information for each machinable polygon.

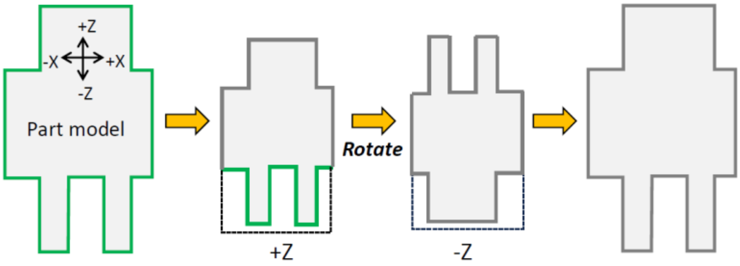

Figure 10: Detection results for minimum workpiece posture changes.

The following is a detailed description of the proposed implementation method. In conventional software, only *machined* or *unmachined* status is recorded for each small polygon. The new implementation expands this recording area to record *machined* (= 1) or *unmachined* (= 0) status for each of the +Z, +X, -X, +Y, -Y, and -Z workpiece postures for each polygon. In the initialization of **Step 1**, an *unmachined* mark (0) is recorded for the six postures of all polygons. In **Step 2.3**, the machined or unmachined status of each polygon is checked, and if it is machined, the area corresponding to the workpiece posture of that polygon is set to 1 (*machined*). The process in **Step 3** is also modified to accommodate the new recording method. For each polygon, the six recording areas corresponding to the workpiece postures are checked, and if one area is found to have been set to 1, the polygon is determined to be machinable. If a polygon has 0 in all recording areas, it is judged to be unmachinable, that is, a difficult-to-machine polygon.

Finally, based on the data recorded in the polygons, the following algorithm is used to determine the required combination of workpiece postures.

Step 1: Examine all small polygons and extract the polygons with zeros recorded in all six recording areas. These polygons are excluded from further processing because they are difficult to machine in any workpiece posture.

Step 2: For each combination of the six workpiece postures (one-posture combination: 6; two-posture combination: 15; three-posture combination: 20; four-posture combination: 15; five-posture combination: 6), operations are performed as follows:

- For all polygons, the recording area corresponding to a given posture combination is examined. If 1 is recorded for any of the recording areas, the polygon can be processed for the given posture combination. If all polygons are detected as machinable, the workpiece posture combination is recorded as the required workpiece posture.

Step 3: The smallest combination of recorded workpiece postures is output as a solution.

Because there are only 62 (= 6 + 15 + 20 + 15 + 6) possible combinations when all combinations are examined, a small increase in processing time is required compared with the conventional method. The method proposed in this study is considered effective because there are at most six possible workpiece postures. As the number of possible workpiece postures increases, the number of posture combinations increases exponentially, making it difficult to use.

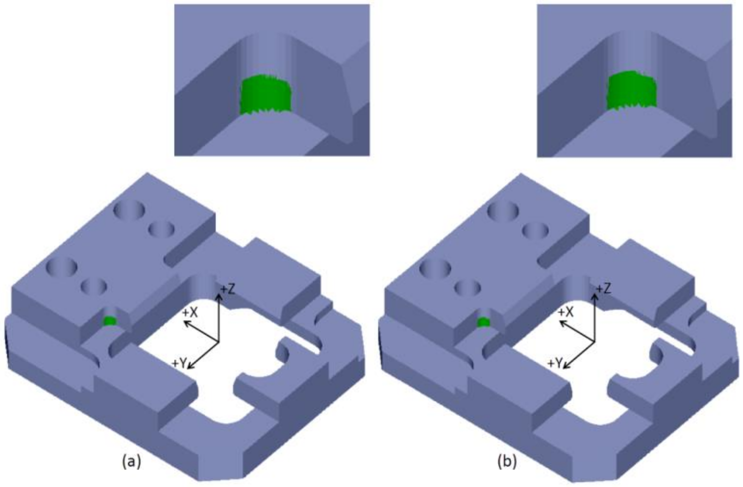

Figure 11: Machining simulation result with six workpiece postures (+Z, +X, -X, +Y, -Y, -Z) (a) and another result obtained with the minimum workpiece postures (+Z, +Y, -Y, -Z) calculated by our system (b).

5 COMPUTATIONAL EXPERIMENTS

The algorithm just described was implemented using Visual Studio 2017 to detect difficult-to-machine shapes for 20-part models provided by a company, calculating the minimum combination of workpiece postures required. The increase in computation time due to additional processing was negligible (tens of milliseconds). Cutting simulations were then performed only for the workpiece posture combinations thus obtained. The simulation results were then compared with those obtained using the conventional method that performs cutting simulations for all six postures. The results were in perfect agreement for all 20 cases, thereby validating the effectiveness of the proposed method. Fig. 11 shows an example of cutting simulation results. Fig. 11(a) shows the results obtained by cutting simulation for six postures (+Z, +X, -X, +Y, -Y, -Z axis directions), and (b) shows the results which were obtained by performing cutting simulations for the limited postures (+Z, +Y, -Y, -Z axis directions) calculated by the proposed software. The results are consistent in both cases.

6 CONCLUSIONS AND FUTURE WORKS

This study describes a software system that determines the minimum combination of necessary workpiece postures for cutting a machine part by extending the previous software to detect difficult-to-machine shapes. Our previous system detected difficult-to-machine shapes by repeating cutting simulations while changing the posture of the workpiece model. The newly developed technology extends this technique to automatically determine the necessary workpiece postures by appropriately using the simulation results. During preparation, the surface of the part model is converted into a set of fine polygons. Considering six different workpiece postures when cutting a mechanical part is rather common. Geometric cutting simulations were performed for each posture, to record the possible workpiece postures for machining each polygon in the part model. Finally, the combinations of workpiece postures that allowed machining of all polygons were examined, whereby the smallest combination was output as the calculation result. Computational experiments were conducted on several part models confirming that appropriate combinations of workpiece postures can be obtained. The time required for this newly added processing was negligible.

Finally, possible future enhancements of this method are presented.

(1) In the proposed system, the required workpiece postures were selected from six predefined postures. This method should be extended to automatically determine candidate postures based on the part geometry without expecting predefined postures. This problem is similar to determining the necessary cutter directions in 3+2 axis machining, requiring the development of a method that considers the aforementioned differences between five- and three-axis machining for mechanical part manufacturing.

(2) The number of tools used in machining also has a significant impact on the machining cost. The current software performs machining simulations using all available tools, with some not contributing to workpiece removal. Thus, software that determines the necessary tool set should be developed.

Our software does not consider the influence of the workpiece shape in the cutting simulation. To improve the accuracy of detecting difficult-to-machine shapes and determine the necessary workpiece postures, an algorithm that preferably considers the workpiece shape in the solution should be implemented. This method would make calculation much more difficult because the workpiece shape changes depending on the order of tools used and workpiece posture changes. The authors believe that introducing a technology that prioritizes a more promising order of posture change and tool selection is necessary.

Masatomo Inui, https://orcid.org/0000-0002-1496-7680
Sunao Kawai, https://orcid.org/0009-0001-3934-7701
Nobuyuki Umezu, https://orcid.org/0000-0002-7873-7833

REFERENCES

[1] Al-wswasi, M.; Ivanov, A.; Makatsoris, H.: A survey on smart, automated computer-aided process planning (ACAPP) techniques, The International Journal of Advanced Manufacturing Technology, 97, 2018, 809–832. https://doi.org/10.1007/s00170-018-1966-1

[2] Bohez, E.L.J.; Senadhera, S.D.R.; Pole, K.; Duflou, J.R.; Tar, T.: A geometric modeling and five-axis machining algorithm for centrifugal impellers, Journal of Manufacturing Systems, 16(6), 1997, 422–436. https://doi.org/10.1016/S0278-6125(97)81700-1

[3] Chuang, S.H.; Henderson, M.R.: Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs, Computer-Aided Design, 22(6), 1990, 377–387. https://doi.org/10.1016/0010-4485(90)90088-T

[4] Farouki, R.T.; Li, S.: Optimal tool orientation control for 5-axis CNC milling with ball-end cutters, Computer Aided Geometric Design, 30(2), 2013, 226–239. https://doi.org/10.1016/j.cagd.2012.11.003

[5] Gavankar, P.; Henderson, M.R.: Graph-based extraction of protrusions and depressions from boundary representations, Computer-Aided Design, 22(7), 1990, 442–450. https://doi.org/10.1016/0010-4485(90)90109-P

[6] Inui, M.: Fast inverse offset computation using polygon rendering hardware, Computer-Aided Design, 35(2), 2003, 191–201. https://doi.org/10.1016/S0010-4485(02)00052-0

[7] Inui, M.; Ohta, A.: Using a GPU to Accelerate Die and Mold Fabrication, IEEE Computer Graphics and Applications, 27(1), 2007, 82-88. https://doi.org/10.1109/MCG.2007.23

[8] Inui, M.; Umezu, N.: Quad Pillars and Delta Pillars: Algorithms for Converting Dexel Models to Polyhedral Models, ASME Journal of Computing and Information Science Engineering, 17(3), 2017, 031001 (9 pages). https://doi.org/10.1115/1.4034737
[9] Inui, M.; Zhang, T.; Umezu, N.: Milling Simulation-Based Method to Evaluate Manufacturability of Machine Parts, Proceedings of ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE2020, August 16-19, 2020, St. Louis, MO, USA, IDETC IDETC2020-22124. https://doi.org/10.1115/DETC2020-22124
[10] Inui, M.; Chen, Q.; Umezu, N.: Improved Method for Extracting Difficult-To-Machine Shapes Using Multiple Milling Simulation Results, Proceedings of the ASME 2022, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE2022, August 14-17, 2022, St. Louis, Missouri, DETC2022-89579. https://doi.org/10.1115/DETC2022-89579
[11] Joshi, S.; Chang, T.C.: Graph-based heuristics for recognition of machined features from a 3D solid model, Computer-Aided Design, 20(2), 1988, 58–66. https://doi.org/10.1016/0010-4485(88)90050-4
[12] Jousselin, B.; Quinsat, Y.; Tournier, C.: A 5-axis pocket roughing strategy reducing the remaining material volume, Procedia CIRP, 82, 2019, 368–373. https://doi.org/10.1016/j.procir.2019.04.146
[13] Kim, Y.S.: Recognition of form features using convex decomposition, Computer-Aided Design, 24(9), 1992, 461–476. https://doi.org/10.1016/0010-4485(92)90027-8
[14] Makhanov, S.S.: Adaptable geometric patterns for five-axis machining: a survey, The International Journal of Advanced Manufacturing Technology, 47, 2010, 1167–1208. https://doi.org/10.1007/s00170-009-2244-z
[15] Misumi, Meviy; https://meviy.misumi-ec.com/en_us-us/
[16] Rao, M.; Ismail, F.; Bedi, S.: Tool path planning for five-axis machining using the principal axis method, International Journal of Machine Tools and Manufacture, 37(7), 1997, 1025–1040. https://doi.org/10.1016/S0890-6955(96)00046-6
[17] Sakurai, H.; Dave, P.: Volume decomposition and feature recognition, Part II: Curved objects, Computer-Aided Design, 28(6–7), 1996, 519–537. https://doi.org/10.1016/0010-4485(95)00067-4
[18] Tang, K.; Woo, T.: Algorithmic aspects of alternating sum of volumes. Part 1: Data structure and difference operation, Computer-Aided Design, 23(5), 1991, 357–366. https://doi.org/10.1016/0010-4485(91)90029-V
[19] Woo, Y.: Fast cell-based decomposition and applications to solid modeling, Computer-Aided Design, 35(11), 2003, 969–977. https://doi.org/10.1016/S0010-4485(02)00144-6
[20] Yeo, C.; Kim, B.C.; Cheon, S.; Lee, J.; Mun, D.: Machining feature recognition based on deep neural networks to support tight integration with 3D CAD systems, Scientific Reports, 11, 2021, 22147. https://doi.org/10.1038/s41598-021-01313-3
[21] Zhu, J; Kato, M.; Tanaka, T.; Yoshioka, H.; Saito, Y.: Graph-based automatic process planning system for multi-tasking machine, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 9(3), 2025. https://doi.org/10.1299/jamdsm.2015jamdsm0034