Automated Process Planning System for Machining Injection Molding Dies Using CAD Models of Product Shapes in STL Format

Nishida, Isamu

Yamada, Eiki

Nakatsuji, Hidenori

(Citation)
International Journal of Automation Technology,17(6):619-626

(Issue Date)
2023-11-05

(Resource Type)
journal article

(Version)
Version of Record

https://doi.org/10.20965/ijat.2023.p0619

Research Paper:

Automated Process Planning System for Machining Injection Molding Dies Using CAD Models of Product Shapes in STL Format

Isamu Nishida*^,†^, Eiki Yamada, and Hidenori Nakatsuji***

*Kobe University
1-1 Rokko-dai, Nada-ku, Kobe, Hyogo 657-8501, Japan
†Corresponding author, E-mail: nishida@mech.kobe-u.ac.jp
**Trend Micro Incorporated, Tokyo, Japan
[Received April 10, 2023; accepted June 12, 2023]

In this study, we developed a method for automatically generating computer-aided design (CAD) models of injection molding dies. The method only required 3D CAD models of products in the Standard Triangulated Language (STL) format as the input information. We also developed a system for automatically generating numerical control (NC) programs by automating the system process planning necessary for machining the injection molding dies. The method generated CAD models of the injection molding dies by dividing the STL files of the products into triangular meshes on a specified split plane. For injection molding dies with several free curved surfaces, we acquired the tool positions of a ball end mill (as approximated by a spherical shape) and flat drill (as approximated by a cylindrical shape) from the geometrical relationships of the triangles constituting the CAD model. We generated a CAD model of an injection molding die using the proposed method with respect to the CAD model of a product shape to verify the validity of the developed system. Then, we machined the product based on the NC programs and tool position. In addition, we injection molded a product with a machined die to mold it into its original product shape.

Keywords: CAM, STL format, injection molding dies, end milling

1. Introduction

Recently, Japan's manufacturing industry has faced challenges regarding its young generation (owing to a declining birthrate) and aging population (from decreases in employees owing to the retirement of skilled laborers). Nevertheless, consumers are demanding customized production systems owing to the diversification of their needs. As a result, the balance between supply and demand has collapsed. Correspondingly, Japan must improve its production efficiency while compensating for the above-mentioned deficiencies in human resources at production sites. Measures for improving production efficiency include production planning and the optimization of supply chains; in this context, it is extremely effective to automate the production process planning in factories for small-lot productions of multiple products.

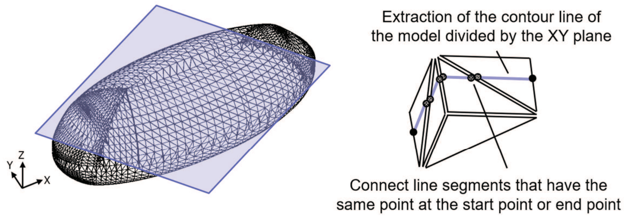
Numerical control (NC) machine tools for automating machining have become prevalent worldwide. For example, end milling is essential for machining complex-shaped parts and free curved surfaces and is employed at many manufacturing sites. As NC programs are essential for conducting automated machining operations with NC machine tools, software has rapidly developed to support their NC programming, such as computer-aided design (CAD)/computer-aided manufacturing (CAM) approaches.

To date, injection molding approaches have been used to mass-produce products and dies, particularly in cases of resin products. Although dies, once fabricated, can be used to mass-produce products, separate dies must be fabricated for individual products. This requires individual setup plans, as well as time and efforts for their machining preparations. In addition, the use of dies is generally positioned between the development designs and start of mass productions; as such, they need to be delivered in a short period of time. However, as the dies constitute the matrixes of the end products, they must be highly accurate and trial-cut, making it difficult to shorten their lead times. Especially when developing new products, although CAD models of the end products are generated during the development process, in many cases, CAD models of the dies are not. Moreover, as the life cycles of products are shortened, the intervals for fabricating their dies have also become significantly shorter. Although the machining of dies represents a typical small-lot or single-item production, the production efficiency is required to match that of mass productions. In the above-mentioned context, many studies have been conducted on design or process planning for the machining of dies. For example, Harada et al. [1], Park and Lee [2], Koresawa et al. [3], and Murata et al. [4] proposed design methods for molds. Nee et al. [5] proposed a method for determining the optimal parting directions in a plastic injection mold design. Guo [6] and Sawa [7] proposed simulation and automated methods for mold and die grinding, respectively. Koresawa et al. [8] studied the use of additive manufacturing for creating an injection mold. Takizawa et al. [9] and Hashimoto and Nakamoto [10] proposed methods for estimating mold machining times and recognizing mold machining patterns using artificial intelligence and deep learning, respectively. However, few studies are available on generating CAD models of dies by using the CAD models of products as input information or on automatically generating the NC programs for their machining. Conventionally, the CAD model of the mold has been created by using the cavity function in the CAD software to create a CAD model of the product. Therefore, the CAD model of the mold cannot be created without the same CAD software that created the CAD model of the product, leading to problems with data compatibility.

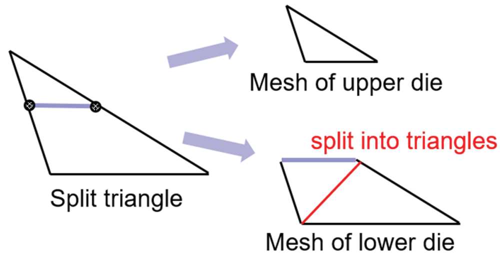
Accordingly, in this study, we developed a system for generating CAD models of injection molding dies using only CAD models in the Standard Triangulated Language (STL) format. The STL format expresses 3D shapes using only triangular meshes of products. This allows users to overcome data compatibility issues, to automate the process planning necessary for machining, and to automate NC programming. Thus, this study proposes a novel method for creating a mold model using only a CAD model of a product shape in the STL format. It also proposes a novel method for performing parallel processing on a graphics processing unit (GPU) based on a geometric calculation of spheres or cylinders and triangular meshes, using only CAD models in the STL format composed of complex free-form surfaces. Thus, this study facilitates the fabrication of dies (which used to be only cost-effective for mass productions) effective for medium-volume productions.

2. Transformation from CAD Models of Product Shapes in STL Format to CAD Models of Injection Molding Dies

Recently, the diversity of the 3D CAD model format has become one of the major problems in realizing automated systems. Various CAD software packages are used in the field. Owing to the differences in these data formats, the 3D CAD data designed by an ordering source are often manually re-created using different CAD software at the manufacturing site, albeit with data compatibility. The STL format expresses 3D shapes using only triangular meshes. The STL format is supported by most CAD software packages owing to its simplicity of the data format. Therefore, we focus on the STL format to overcome data compatibility issues. In this study, we propose a method for generating CAD models of injection molding dies by dividing the CAD models of product shapes in the STL format on an arbitrary plane. Taking the 3D CAD model of the product shape shown in **Fig. 1(a)** for example, we can elaborate on how to generate the CAD



(a) Contour line obtained by target CAD model and split plane



(b) Triangular mesh divided by XY plane

Fig. 1. Division of 3D CAD model of product.

model of its injection molding die. In this study, the split plane necessary for generating the die model is given as the input information. As shown in **Fig. 1(a)**, we identify the contour lines as acquired by splitting the model on the specified XY plane. In this study's target CAD model in the STL format, the surface of the product shape is composed of triangular meshes. As such, we first extract the triangular mesh straddling the Z-coordinate on the target's XY plane. Next, we divide the extracted triangular mesh on the target's XY plane as shown in **Fig. 1(b)**, by which one triangle and one trapezoid are extracted. By dividing the extracted trapezoid by the straight line connecting the vertex on the one side of its upper edge to the vertex on the one side of its lower edge, we can newly extract two triangles. We divide the extracted trapezoid by a straight line in such a way that the interior angle of each vertex of the newly extracted triangles ideally becomes as large as possible. If the Z-coordinate of any one of the vertexes of the product shape's triangular mesh is equal to the Z-coordinate of the split plane, we do not divide the triangular mesh. The above-mentioned analysis enables us to divide the 3D CAD model of a product shape into a triangular mesh necessary for the upper die and into one necessary for the lower die. Then, to generate the die model, we add a mesh containing the extracted triangular meshes. For example, in generating the lower die model, we only extract the triangular mesh present in the lower part of the split plane and connect it to the rectangular parallelepiped containing the extracted triangular mesh, as shown in **Fig. 2**. As the normal vectors of the triangular meshes extracted from the CAD model of a product shape are directed outward with reference to the product shape, we need to invert them to generate the CAD model of the die model in the STL format. The above-mentioned processing enables us to generate the CAD model of the die by specifying an arbitrary split plane of the CAD model of the product shape, as shown in **Fig. 3**.

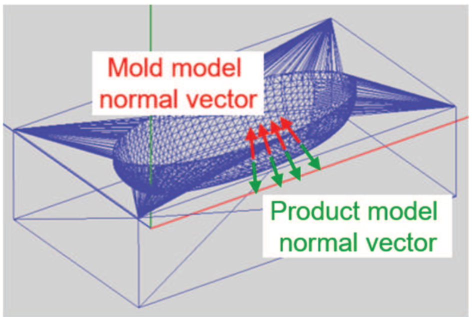


Fig. 2. Generation of mold model with divided triangular mesh.

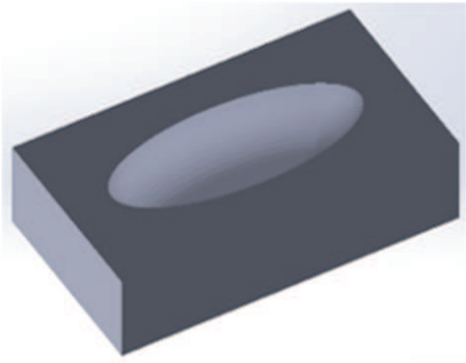


Fig. 3. Calculation of tool paths for end milling injection molding dies.

3. Calculation of Tool Paths for End Milling Injection Molding Dies

Many previous studies have been conducted on process planning, including those on recognition of a machining feature from the target product shape [11–14] and that from a removal volume [15–18]. These studies also proposed methods for calculating the tool path for the contour machining from the extracted machining feature [19].

As an injection molding die model often contains inclined and free curved surfaces on its component surface, it is difficult to extract the machining features from the CAD model for its process planning and to contour-machine it as conventionally performed. In this study, therefore, we conduct scanning from the CAD model of the product shape in the STL format. Scanning refers to a method of machining along the surface of a product shape in either a zigzag or one-way path, as shown in **Fig. 4**. In the scanning, the tool paths constitute straight lines to an arbitrary axis. Thus, we can calculate the tool paths while only considering the offset in the Z-axial direction along the product shape.

In the CAD model in the STL format, the surface of a product shape is expressed by the information of three vertexes and of the normal vectors of a triangle composed of the three vertexes. The use of the information on the triangle's normal vectors enables us to acquire the areas of the inclined and curved surfaces. Insofar as the Z-values of the normal vectors of each triangular mesh, as the normal vector's Z-value 1 or -1 represents a plane and its Z-value 0 represents a side, we can extract a mesh whose Z-value is not -1, 0, or 1 as an inclined or curved surface. Generally, in the injection molding die model, as the areas other than where material is injected to generate a product shape (i.e., the contact surface between the upper and lower dies) are planar, we can leave them as surface-shaped materials. The areas requiring scanning, therefore, are simply the inclined and curved surfaces (recessed areas). The boundary between the recessed area where the material is injected and contact surface is equal to a contour line acquired by dividing a product shape by an arbitrary split plane, as described in Section 2.

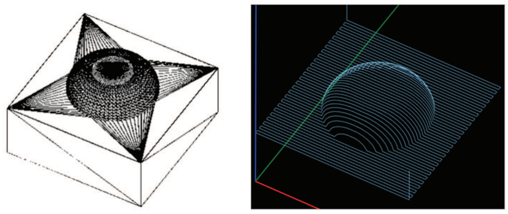


Fig. 4. Example of scanning tool paths (zigzag).

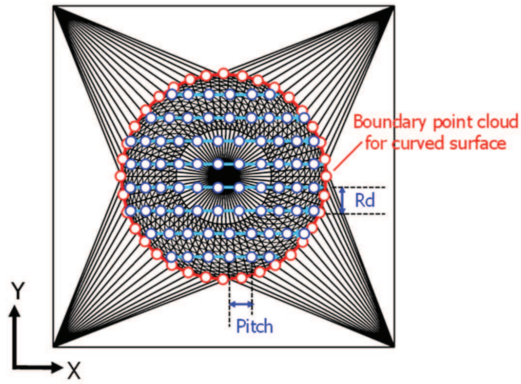
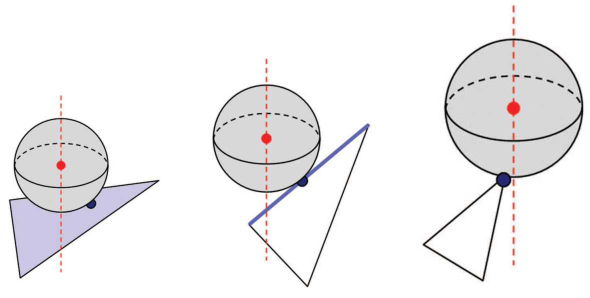


Fig. 5. Tool position on the XY plane in scanning machining.

In the scanning, with respect to the tool position on the XY plane, we can calculate the tool paths while only considering the offset in the Z-axial direction from the interference quantity between the product shape and tool. First, we seek the tool position on the XY plane from the tool's radial depth of cut (Rd) and the resolution (Pitch) in the tool's moving direction, as shown in **Fig. 5**.

Next, we seek the offset in the Z-direction for each of the tool positions determined on the XY plane. This study describes a geometrical analysis of ball end milling. In the case of ball end milling, we can set the position where the ball in the ball part at the end contacts the triangular mesh as an offset. With respect to all of the triangular meshes, we geometrically determine the sphere's center position that generates the largest Z-value among the positions where the sphere contacts the plane (**Fig. 6(a)**), where the sphere contacts the edge (**Fig. 6(b)**), and where the sphere contacts the vertex (**Fig. 6(c)**).



(a) Plane (b) Edge (b) Vertex

Fig. 6. Positional relationship between tool and triangular mesh contacts.

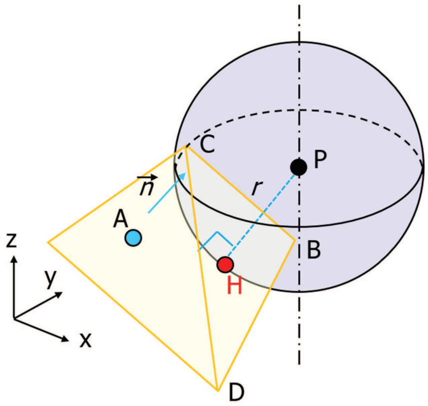


Fig. 7. Geometric relationship between sphere and plane.

Figure 7 shows the positional relationship between the sphere and plane. In the 3D space, with the unit normal vector denoted by $\vec{n}$ and the point where Point P is projected on the plane represented by the Point A that passes there, by H, the relationships are as follows:

$$H = P - \left(\vec{n} \cdot \overrightarrow{AP}\right)\vec{n} \quad \text{(1)}$$

As the radius of the sphere with its center positioned at Point P corresponds to the tool's radius r, a corresponding calculation can be made as follows:

$$r = \left|\overrightarrow{PH}\right| \quad \text{(2)}$$

Furthermore, as the point of contact with the sphere needs to exist within the target's triangular mesh, an equation can be constructed as follows:

$$\overrightarrow{BH} = s\overrightarrow{BD} + (1-s)\overrightarrow{BC} \quad \text{(3)}$$
$$0 < s < 1$$

Figure 8 shows the positional relationship between the edge and sphere. In the 3D space, with the point where the perpendicular line from Point P intersects the line segment EF denoted by I, the relationship is as follows:

$$\overrightarrow{IP} \cdot \overrightarrow{EF} = 0 \quad \text{(4)}$$

As the radius of the sphere with its center positioned at Point P corresponds to the tool's radius r, r can be expressed as follows:

$$r = \left|\overrightarrow{PI}\right| \quad \text{(5)}$$

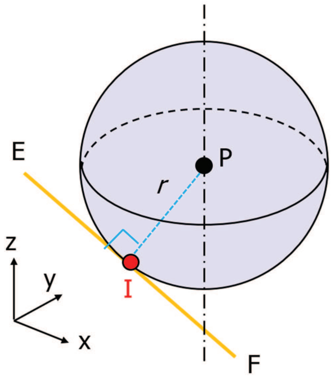


Fig. 8. Geometric positional relationship between spheres and edges.

Furthermore, as the point of contact with the sphere needs to exist on the line segment EF, corresponding calculations can be constructed as follows:

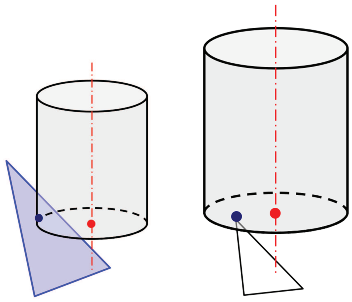
$$\overrightarrow{PI} = u\overrightarrow{PE} + v\overrightarrow{PF} \quad \text{(6)}$$
$$0 < u < 1$$
$$0 < v < 1$$
$$u + v = 1$$

The positional relationship of the point where the sphere contacts the vertex is included in the positional relationship of the point where the sphere contacts the edge.

We calculate each of the tool positions determined on the XY plane by seeking the coordinate of Point P with the largest Z-value satisfying the above-mentioned Eqs. (1)–(6) among all of the triangular meshes and their edges.

The above-mentioned procedures enable us to generate the tool paths for finishing the inclined and curved surfaces of a product shape.

For the roughing, we use a flat drill, as proposed by another research group [20]. In ball end milling or square end milling, the tool's side-cutting edge is generally used. Although there is a cutting edge at the end of a ball end mill, its peripheral speed approaches zero, making it impractical to use in machining. Moreover, as no cutting edge exists on the bottom face of a square end mill, we have to avoid feeding the tool, (whether a ball end mill or square end mill) to its axial direction for any material. Although a flat drill is planar at the end and thus cannot be laterally fed for machining unlike in an ordinary end mill, it is suited for thrust cutting. In the case of a flat drill, we can determine the tool position by considering the position where the cylinder in the tool shape contacts the triangular mesh in the product shape as an offset in the Z-direction. In seeking the flat drill's tool position, with respect to all of the triangular meshes, we geometrically determine the center position of the cylinder's



(a) Plane (b) Vertex

Fig. 9. Positional relationship between the cylinder and point of contact of the triangular mesh.

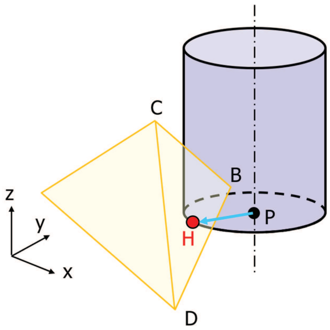


Fig. 10. Geometric positional relationship between planes and cylinders.

bottom face where the Z-value becomes the largest at either of the points where the cylinder contacts the plane or edge (**Fig. 9(a)**) and where the cylinder contacts the vertex (**Fig. 9(b)**).

Figure 10 shows the positional relationship of the cylinder contacting the plane or edge. With the center of the cylinder's bottom face denoted by P and its radius by R, its contact Point H with the triangle BCD can be expressed as follows:

$$R = \left|\overrightarrow{PH}\right| \qquad (7)$$

$$\begin{aligned}\overrightarrow{BH} &= m\overrightarrow{BC} + n\overrightarrow{BD}\\ 0 &< m < 1\\ 0 &< n < 1\\ 0 &< m+n < 1\end{aligned}$$

Furthermore, the contact Point H exists somewhere on the intersection line between the plane represented by the cylinder bottom face and that represented by the triangle BCD. As shown in **Fig. 11**, we denote the intersection line between Planes $F1$ and $F2$ by L.

With Plane 1 expressed by Eq. (1) and Plane 2 by Eq. (2), we determine the intersection line L between the two planes from Eq. (3) using Point H where the intersection line passes, the intersection line's direction vectors $\vec{e}(e_x, e_y, e_z)$, and parameter t.

$$a_1x + b_1y + c_1z = d_1 \qquad (8)$$

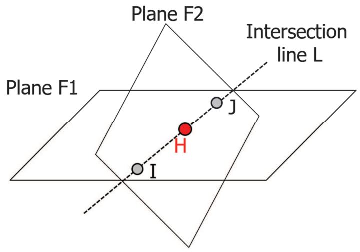


Fig. 11. Relationship between two planes and their lines of intersection.

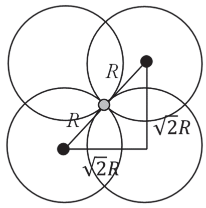


Fig. 12. Interval between the centers of the holes.

$$a_2x + b_2y + c_2z = d_2 \qquad (9)$$

The intersection line's direction vectors $\vec{e}$ constitute the normal and perpendicular vectors of the two planes and can be acquired using exterior product calculations. The components of the direction vectors can be expressed as follows:

$$\begin{aligned}e_x &= b_1c_2 - c_1b_2\\ e_y &= c_1a_2 - a_1c_2\\ e_z &= a_1b_2 - b_1a_2\end{aligned} \qquad (10)$$

As we only consider three-axial machining herein, $e_z = 0$. With two arbitrary points on the intersection line denoted by I and J, using the intersection line's direction vector $\vec{e}$, an equation can be constructed as follows:

$$\overrightarrow{IJ} = t\vec{e} \qquad (11)$$

Furthermore, as the contact point H exists on the intersection line L, the corresponding calculation is as follows:

$$\overrightarrow{IH} = w\overrightarrow{IJ} \qquad (12)$$

We can determine the contact point H between the target's triangular mesh and flat drill's circular bottom face using the above-mentioned relational expressions.

In the case where the triangle's vertex contacts the cylinder's bottom face, the Z-value of the bottom face becomes equal to that of the vertex where the triangle contacts the cylinder's bottom face. With respect to the tool positions determined on the XY plane, we calculate the flat drilling machining tool position by seeking the coordinate of Point P where the Z-value satisfying Eqs. (7)–(12) becomes the largest among all of the triangular meshes.

As a flat drill is planar at the end, it can perform overlapping hole-drilling, unlike an ordinary drill with a sharp end. As shown in **Fig. 12**, therefore, we conduct roughing with a flat drill by machining overlapping holes with a certain space between them. The space between the overlapping holes in the Y-axial direction is equal to the radial depth of the cut to be specified. If the space between the overlapping holes in the X-axial direction is 0.7 or less times the radius of the tool to be used, we can rough machine with it, leaving no areas unmachined except at the edge parts.

With respect to each of the tool positions determined on the XY plane, we need to determine the tool's Z-value using the above-mentioned geometrical calculations. This is because in cases of finishing where machining accuracy is required, Rd and Pitch in the tool's moving direction may be so small in value that the number of points to be determined becomes enormously large. Moreover, in the case of an injection molding die (i.e., the object of this study), the number of meshes of the CAD models as an analysis object can become so large that it takes a massive processing time to determine the tool position, correspondingly lengthening the analysis time. Therefore, we shorten the analysis time by using a GPU for parallel processing. The tool position determination processing at each XY-position can be performed independently and is identical to the processing to determine the point of contact between the sphere and triangular mesh. GPUs were originally intended for use for image processing; here, we apply one to the analyses. We need to transfer the memory information of the CPU as a main memory to the GPU for its parallel processing and then to transfer the processing results to the CPU. Although such data-transferring processing constitutes overhead, it can increase the amount of parallel processing: the longer time one processing requires, the more effective it becomes. Although the specific designs for transferring information from CPU to GPU are important, we perform the parallel processing by (1) transferring the vertexes and normal vectors of all triangular meshes of the CAD model as a processing object, (2) transferring the X- and Y-coordinates of each point and tool diameters based on determining the tool's Z-position at each GPU core by the method described in Section 2, and (3) transferring the processing results to the CPU. We can significantly shorten the analysis time by transferring the information on the vertex coordinates and normal vectors of all of the triangular meshes, X- and Y-values of the tool positions determined on the XY plane, and the tool radii to the GPU memory. Thus, the processing for determining the Z-value at each point can be executed in each GPU core.

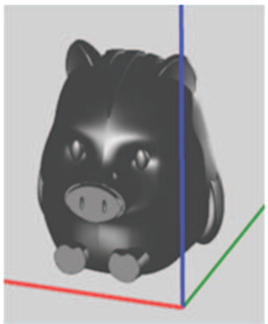


Fig. 13. CAD model used for case study.

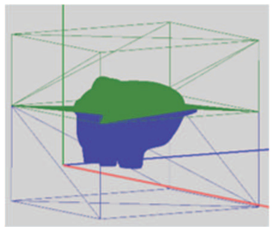


Fig. 14. Triangular mesh of generated mold.

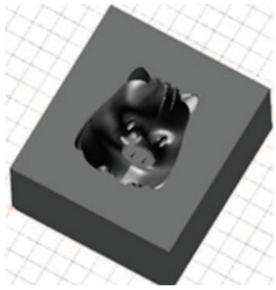


Fig. 15. CAD model of generated mold (lower mold).

Table 1. Machining conditions used in the case study.

(a) Roughing with a flat drill

Tool type	Flat drill
Tool diameter	4.0 mm
Radial depth of cut	0.5 mm
Spindle speed	6000 min^{-1}
Feed rate	100 mm/min

(b) Finishing with a ball end mill

Tool type	Ball end mill
Tool diameter	1.0 mm
Radial depth of cut	0.02 mm
Feed direction resolution (tolerance)	0.02 mm
Spindle speed	10000 min^{-1}
Feed rate	200 mm/min

4. Case Study

To verify the validity of the proposed method, we conducted analyses using the STL data of the CAD model of a product shape, as shown in **Fig. 13**. **Fig. 14** shows the CAD model of an injection molding die as generated by the method described in Section 2. In this case study, the position where the normal vector in the Y-axis direction of the CAD model switched between positive and negative was obtained in advance and used as the split plane. **Fig. 15** shows the CAD model as drawn while considering the natural light reflection to prove that the normal vectors are correctly directed. The tool paths for the lower die model are machined based on the conditions described in **Table 1**. **Fig. 16** shows the toolpaths for roughing with a flat drill as acquired by the proposed method. **Fig. 17** shows the toolpaths for finishing with a ball end mill. In this case study, after roughing the model with a flat model, prior to its finishing, we apply semi-finishing to the toolpaths for the finish scanning by diminishing the machining margin by 0.1 mm stepwise from 0.5 mm. We used

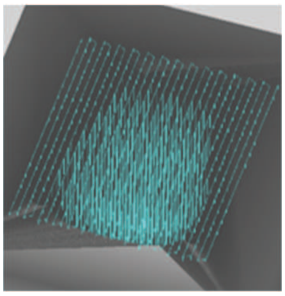


Fig. 16. Toolpath for roughing with a flat drill.

Fig. 17. Toolpath for finishing with a ball end mill.

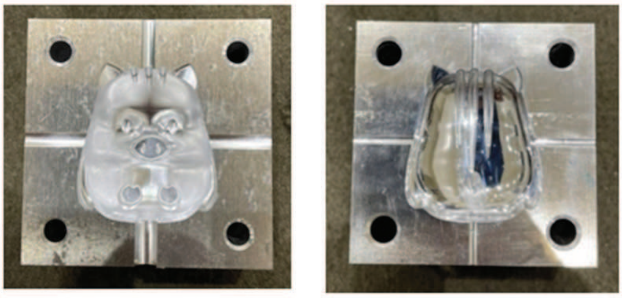


Fig. 18. Machined results.

Fig. 19. Injection molding result with machined mold.

the Intel Core i7-1065 as the CPU and NVIDIA GeForce GTX 1650 as the GPU. Based on analyzing the finish machining tool paths, the parallel processing with the GPU requires an analysis time of 3 min, 13 s, whereas the parallel processing without using any GPU requires 99 min, 51 s. **Fig. 18** shows a workpiece (aluminum alloy A5052) machined with an NC machine tool Robodrill α-D14MiB (made by Fanuc Corporation). We can see from the machined workpiece in **Fig. 18** that there are no problems in the generated tool paths.

Finally, using the machined injection molding die, we injection molded a polypropylene material with an injection molding machine (MoldLock X-801). **Fig. 19** shows the injection molding results. The purpose of this study is to automatically generate a CAD model of a mold from a CAD model in the STL format of a product shape including a free-form surface; as such, it is difficult to evaluate the accuracy after injection. In this study, we assume that the accuracy of the mold depends on the machining conditions for the finish cutting. A quantitative evaluation of the injection molding results will be conducted as a future research topic. The use of the system constructed in this study enables us to significantly shorten the lead times for the injection molding production of products from 3D CAD models expressed in the STL format.

5. Conclusions

In this study, we proposed a method for automatically generating CAD models of injection molding dies, using only the information of the 3D CAD models of the products as inputs. We also developed a system for automatically generating NC programs by automating the process planning necessary for machining the generated injection molding dies. The conclusions of this study are summarized below.

1. We generated the CAD model of an injection molding die by dividing the CAD model of a product expressed in the STL format into triangular meshes on a specified split plane, inverting the normal vectors of the triangular meshes acquired by the abovementioned division, and adding a mesh to provide the outer frame of the injection molding die.

2. With respect to a CAD model in the STL format having many free curved surfaces, we generated its scanning tool paths from the geometrical relationships between the sphere/cylinder and triangle. Furthermore, we reduced the calculation time by employing parallel processing with a GPU.

3. We verified the proposed method through actual machining to confirm that we could machine injection molding dies without problems using the NC programs generated by the proposed method. The results from the injection molding with the machined die were valid.

In our approach, we need to specify the split plane of the CAD model of a product in advance to extract the CAD model of its injection molding die. To further promote the automated generation of injection molding dies, we will study a method for automatically determining the positions of split planes and number of divisions from the features of the triangular meshes constituting the CAD models of products.

Acknowledgments

This work was partially supported by JSPS KAKENHI (Grant Number JP22K03862).

References:

[1] T. Harada, N. Tanaka, and T. Fujitsuka, "Design of an Arc-Core Moving Mechanism for Injection Molding Using a Link and Cam Mechanism," Int. J. Automation Technol., Vol.15, No.3, pp. 366-374, 2021. https://doi.org/10.20965/ijat.2021.p0366

[2] J. C. Park and K. Lee, "Computer Aided Design of a Mold Cavity With Proper Rigging System for Casting Processes: Part 2," J. of Manufacturing Science and Engineering, Vol.113, Issue 1, 1991. https://doi.org/10.1115/1.2899624

[3] H. Koresawa, H. Fukumaru, M. Kojima, J. Iwanaga, H. Narahara, and H. Suzuki, "Design Method for Inner Structure of Injection Mold Fabricated by Metal Laser Sintering," Int. J. Automation Technol., Vol.6, No.5, pp. 584-590, 2012. https://doi.org/10.20965/ijat.2012.p0584

[4] Y. Murata, H. Suzuki, and S. Kashiwagi, "Development of an Injection Mold Capable of Melt Flow Control and Induction Heating and Cooling," Int. J. Automation Technol., Vol.11, No.6, pp. 985-992, 2017. https://doi.org/10.20965/ijat.2017.p0985

[5] A. C. Nee, M. W. Fu, J. Y. H. Fu, K. S. Lee, and Y. F. Zhang, "Determination of Optimal Parting Directions in Plastic Injection Mold Design," CIRP Annals, Vol.46, Issue 1, pp. 429-432, 1997. https://doi.org/10.1016/S0007-8506(07)60858-0

[6] C. Guo, "Modeling and Simulation of Mold and Die Grinding, J. of Manufacturing Science and Engineering," Vol.134, Issue 4, 2012. https://doi.org/10.1115/1.4006970

[7] T. Sawa, "Automating the Mold-Material Grinding Process," Int. J. Automation Technol., Vol.13, No.6, pp. 722-727, 2019. https://doi.org/10.20965/ijat.2019.p0722

[8] H. Koresawa, H. Fujimaru, and H. Narahara, "Improvement in the Permeability Characteristics of Injection Mold Fabricated by Additive Manufacturing and Irradiated by Electron Beams," Int. J. Automation Technol., Vol.11, No.1, pp. 97-103, 2017. https://doi.org/10.20965/ijat.2017.p0097

[9] H. Takizawa, H. Aoyama, and S. C. Won, "Prompt Estimation of Die and Mold Machining Time by AI Without NC Program," Int. J. Automation Technol., Vol.15, No.3, pp. 350-358, 2021. https://doi.org/10.20965/ijat.2021.p0350

[10] M. Hashimoto and K. Nakamoto, "Process planning for die and mold machining based on pattern recognition and deep learning," J. of Advanced Mechanical Design, Systems, and Manufacturing, Vol.15, No.2, 2021. https://doi.org/10.1299/jamdsm.2021jamdsm0015

[11] K. Nakamoto, K. Shirase, H. Wakamatsu, A. Tsumaya, and E. Ara, "Automatic production planning system to achieve flexible direct machining," JSME Int. J. Series C, Vol.47, No.1, pp. 136-143, 2004. https://doi.org/10.1299/jsmec.47.136

[12] L. Wang, M. Holm, and G. Adamson, "Embedding a process plan in function blocks for adaptive machining," CIRP Annals – Manufacturing Technology, Vol.59, Issue 1, pp. 433-436, 2010. https://doi.org/10.1016/j.cirp.2010.03.144

[13] Y. Woo, E. Wang, Y. S. Kim, and H. M. Rho, "A hybrid feature recognizer for machining process planning systems," CIRP Annals – Manufacturing Technology, Vol.54, Issue 1, pp. 397-400, 2005. https://doi.org/10.1016/S0007-8506(07)60131-0

[14] A. Ueno and K. Nakamoto, "Proposal of machining features for CAPP system for multi-tasking machine tools," Trans. of the JSME, Vol.81, No.825, 2015 (in Japanese). https://doi.org/10.1299/transjsme.15-00108

[15] E. Morinaga, M. Yamada, H. Wakamatsu, and E. Arai, "Flexible process planning method for milling," Int. J. Automation Technol., Vol.5, No.5, pp. 700-707, 2011. https://doi.org/10.20965/ijat.2011.p0700

[16] E. Morinaga, T. Hara, H. Joko, H. Wakamatsu, and E. Arai, "Improvement of computational efficiency in flexible computer-aided process planning," Int. J. Automation Technol., Vol.8, No.3, pp. 396-405, 2014. https://doi.org/10.20965/ijat.2014.p0396

[17] H. Sakurai, "Volume decomposition and feature recognition: part 1 – polyhedral objects," Computer-Aided Design, Vol.27, Issue 11, pp. 833-843, 1995. https://doi.org/10.1016/0010-4485(95)00007-0

[18] H. Sakurai and P. Dave, "Volume decomposition and feature recognition, part II- curved objects," Computer-Aided Design, Vol.28, Issues 6-7, pp. 519-537, 1996. https://doi.org/10.1016/0010-4485(95)00067-4

[19] I. Nishida and K. Shirase, "Automated process planning system for end-milling operation by CAD model in STL format," Int. J. Automation Technol., Vol.15, No.2, pp. 149-157, 2021. https://doi.org/10.20965/ijat.2021.p0149

[20] I. Nishida, H. Nakatsuji, and K. Shirase, "Automated tool path generation for roughing using flat drill," Int. J. Automation Technol., Vol.14, No.6, pp. 1036-1044, 2020. https://doi.org/10.20965/ijat.2020.p1036

Name:
Isamu Nishida

ORCID:
0000-0002-3737-2073

Affiliation:
Associate Professor, Department of Mechanical Engineering, Graduate School of Engineering, Kobe University

Address:
1-1, Rokko-dai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
2012- Sysmex Corp.
2016- Assistant Professor, Kobe University
2018- CEO, BESTOWS Co., Ltd. (Concurrently)
2021- CTO, ARUM Co., Ltd. (Concurrently)
2022- Associate Professor, Kobe University

Main Works:

- "Automated process planning system for end-milling operation by CAD model in STL format," Int. J. Automation Technol., Vol.15, No.2, pp. 149-157, 2021.
- "Machining time reduction by tool path modification to eliminate air cutting motion for end milling operation," Int. J. Automation Technol., Vol.14, No.3, pp .459-466, 2020.
- "Sequence planning of on-machine measurement and re-machining," J. of Advanced Mechanical Design, Systems, and Manufacturing, Vol.13, No.1, 2019.

Membership in Academic Societies:

- The Japan Society of Mechanical Engineers (JSME)
- The Japan Society for Precision Engineering (JSPE)

Name:
Eiki Yamada

Affiliation:
Trend Micro Incorporated

Address:
2-1-1 Yoyogi, Shibuya-ku, Tokyo 151-0053, Japan

Brief Biographical History:
2022- Trend Micro Incorporated

Name:
Hidenori Nakatsuji

Affiliation:
Technical Staff, Center of Manufacturing Technology, Graduate School of Engineering, Kobe University

Address:
1-1, Rokko-dai, Nada, Kobe, Hyogo 657-8501, Japan

Brief Biographical History:
2008- Technical Staff, Kobe University

Membership in Academic Societies:

- Society of Automotive Engineers of Japan, Inc. (JSAE)