Hindawi
Advances in Polymer Technology
Volume 2020, Article ID 2075434, 11 pages
https://doi.org/10.1155/2020/2075434

Research Article

Five-Axis Tool Path Generation of Injection Mold Represented by T-Spline Surface

Ce Shang ,[1,2] Hongyao Shen ,[1,2] Jianzhong Fu,[1,2] Yangfan Sun,[1,2] Shuhua Yue,[1,2] and Jianfeng Zhang[1,2]

[1] *State Key Laboratory of Fluid Power and Mechatronic Systems, College of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China*
[2] *Key Laboratory of 3D Printing Process and Equipment of Zhejiang Province, College of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Hongyao Shen; shenhongyao@zju.edu.cn

Received 4 June 2019; Revised 7 August 2019; Accepted 16 August 2019; Published 9 March 2020

Guest Editor: Yun Zhang

Injection molding is widely used in industries to produce polymeric products. At present, compound NURBS surfaces are commonly used to represent freeform surfaces in mold models. This work uses T-spline surface with extraordinary control points instead of NURBS to represent freeform surfaces in mold models. Compared with NURBS, T-splines' higher-order continuity facilitates the mold quality control especially at the patch boundary. And, its patch layout information can be utilized for tool path planning. We propose an algorithm to determine the patch processing order and generate nonretraction tool path for T-spline surface models. The tool paths are generated patch by patch using isoparametric strategy. Actual machining and injection molding experiments have been conducted. The result shows the feasibility of the proposed method, and the final product is in good quality.

1. Introduction

In today's industry, polymer is one of the most widely used materials. To produce polymeric products, injection molding (IM) is extensively used and is characterized by low cost, high precision, high productivity, and production of complex products. In the past few decades, injection molding has achieved rapid development due to the growth of new applications in automotive, transportation, electronics, etc. Currently, about one-third of all polymeric products are produced by injection molding. Many researchers are dedicated to make progress in the field of injection molding. Hot research topics include mathematical modeling [1], online measurement, parameter optimization [2–4], mold design, and manufacturing methods [5–7].

One of the main goals in IM is the improvement of molded product quality. To produce high-quality molded product, a high precision mold has to be provided. As a typical situation of mechanical manufacturing, mold manufacturing also has its particularity. For example, as the geometric shape of a mold is usually complex, mold manufacturing is usually finished by milling operations. And, because the mold sometimes contains deep cavities, the five-axis milling is usually applied to avoid unintended gouges and collision. A lot of research has been reported about mold manufacturing including automatic polishing [8], machining parameter optimization [9, 10], and optimum parting direction determination [11].

Another major characteristic of mold manufacturing is that the mold is usually represented by freeform surfaces. Nonuniform rational B-splines (NURBS) are widely used in the last few decades to represent freeform surfaces for design and manufacturing. To represent a simple shape, a single-patched NURBS surface model might be enough. To represent a complex shape, several NURBS patches are utilized, and each patch might be trimmed as needed. This kind of model is called compound NURBS surface model. Although the compound NURBS surface model is widely applied in CAD and CAM, it still has some disadvantages in use. The redundant control points in NURBS surface increase the required storage space. Cracks or ripples usually appear between adjacent patches. And, the patch layout in compound NURBS surface model is usually not clearly defined. These disadvantages of NURBS trouble the CAD area for decades. And, in CAM area, these disadvantages also make tool path generation difficult. Generally, there are three major strategies to generate a tool path for a freeform surface model [12], isoparamtric [13, 14], isoplanar [15] and isoscallop [16]. The isoparametric strategy uses a series of isoparametric lines as the cutter contact paths. For compound NURBS surface model, isoparametric strategy has to be applied patch by patch as the boundary between adjacent NURBS patches is usually discontinuous. The discontinuity may cause problems such as uncut material at patch boundaries. And, a retraction is usually needed to connect the tool path of two different patches. The machining time might become longer in this situation. The isoplanar strategy uses the intersection lines between the surface model and a series of planes as the cutter contact paths. Although this strategy is less sensitive to the model continuity, it assumes that the direction of the normal vectors of the surface does not change too much over the entire model. Thus, it is not a suitable method when the model has a complex geometry. The isoscallop strategy controls the scallop between two adjacent tool paths to be a constant. It is usually applied patch by patch like the isoparametric strategy. And, it also has the aforementioned shortcoming as the isoparametric strategy.

To overcome the defects of the compound NURBS surface model, T-splines were introduced by Sederberg et al. in 2003 [17]. As a generalization of NURBS, T-splines inherit most of the advantages of NURBS. Besides, T-splines have three attractive features over NURBS. Firstly, T-junctions are allowed in the control mesh of T-splines to avoid redundant control points. The computational cost can be significantly reduced for complex models. Secondly, the preimage of a T-spline surface does not have to be a rectangular region. Holes and irregular boundaries are allowed for a single T-spline patch. Thirdly, T-splines introduce extraordinary control points. Thus, a multipatched model can be merged into a watertight T-spline surface model with high-order continuity [18]. And, the patch layout information is clear in T-splines. With all these advantages, T-spline technique has been regarded as a promising technique. And, it has been regarded as a possible candidate for the future industry standard. In recent years, several works have been reported about T-spline CAM. Gan et al. proposed an improved SFC method to generate a nonretraction tool path for T-spline surfaces with irregular boundary [19]. Zhao et al. developed a STEP-compliant CNC system which uses T-spline surfaces to represent the freeform surface model in the system [20, 21]. And, Feng et al. proposed a T-spline surfaces direct slicing algorithm for FDM 3D printing [22]. Currently, all these existing T-spline CAM research focuses on the situation that the model does not contain extraordinary control points. However, the representation ability of a T-spline surface without extraordinary control point is quite limited. The T-spline surface without an extraordinary control point is no more complex than a single-trimmed NURBS patch. Therefore, T-spline CAM for mold manufacturing has to consider T-spline surface models with extraordinary control points to fully take advantage of T-splines. Thus, the aforementioned works are not suitable in the mold manufacturing application.

In this work, we propose to use T-splines instead of traditional compound NURBS surfaces to represent injection molds. We propose a five-axis tool path generation algorithm facing the T-spline surface model with extraordinary control points. The isoparametric strategy is applied as it is a simple and robust way to control the machining scallop. And, the T-spline patch layout is utilized to generate a nonretraction tool path. The rest of the article is constructed as follows. Section 2 briefly introduces the basic theories of T-splines. Section 3 introduces the proposed algorithm to generate the five-axis tool path. The experimental examples include machining and injection molding are given in Section 4. And, Section 5 summarizes the whole paper and gives the conclusion.

2. Basic Theories of T-Splines

2.1. T-Spline Surface. The basic theory of T-splines can be found in detail in [17]. The algebraic formula of T-spline surface $\mathbf{S}(u,v)$ is the same as it of NURBS, which is defined as follows:

$$\mathbf{S}(u,v)=\frac{\sum_{i=1}^{n} w_i B_i(u,v)\mathbf{P}_i}{\sum_{i=1}^{n} w_i B_i(u,v)}, \tag{1}$$

where $\mathbf{P}_i$ is the i-th control point, w_i is the weight of the control points, n is the number of 96 the control points, $B_i(u,v)$ is the blending function which can be calculated according to the famous de Boor Cox recursion formula while the knot vector is defined using the ray-intersection method in [17]. Figure 1 gives a T-spline surface example, and the enlarged drawing on the right shows the control mesh topology and marked the T-junctions in the control mesh. The existence of T-junctions avoids the redundant control points and also reduces the computational consumption.

2.2. T-Spline Surface with Extraordinary Control Points. A surface with a complex topology cannot be represented using T-splines with a single global u-v parametric representation [23]. Besides, making isoparametric lines align with principal curvature direction will also benefit the shape representation [24]. Extraordinary control points are introduced to improve T-splines from these two perspectives. For T-spline surface with extraordinary control points, the entire surface model is constructed with several T-spline patches. Each patch has its own u-v parametric representation just like a NURBS patch in compound NURBS surface models. And, the parametric representation between the adjacent patches can be transformed with a translation and rotation transform. The basis functions of control points near the patch boundary have C^2 continuity at the patch boundary. Thus, the patch boundary of T-splines has high-

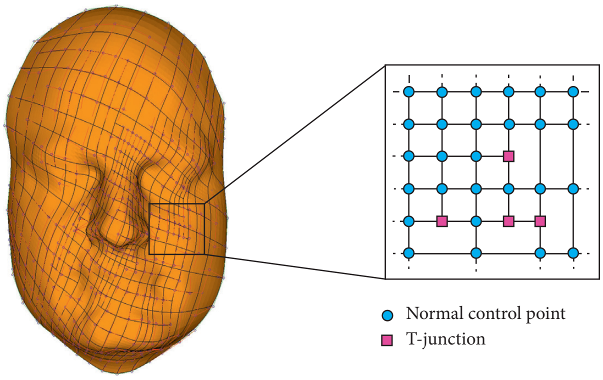


Figure 1: A face model with only one T-spline surface patch. The enlarged drawing shows the T-junctions in the control mesh topology.

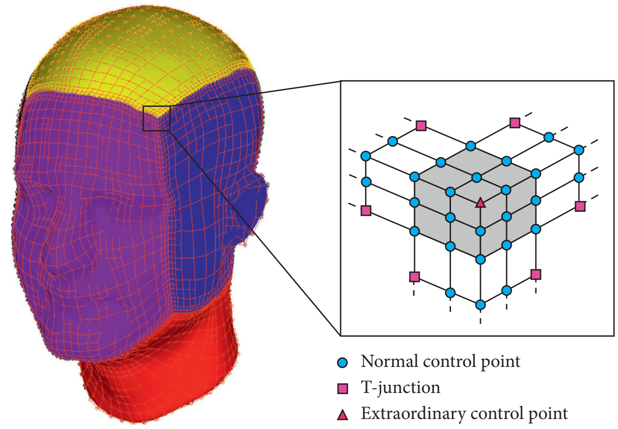


Figure 2: A head model which contains 6 patches and 8 extraordinary control points. Different patches are drawn in different colors. The enlarged drawing shows the extraordinary control point at the patch corner, and the extraordinary region is marked in gray.

order continuity just like the interior of a patch. For the control point at the patch corner, if the number of patches that sharing this corner is not four, this control point is called an extraordinary control point. The region near the extraordinary control point (usually its two-ring neighbor) is called extraordinary region. Figure 2 gives an example of a T-spline surface model with 6 patches and 8 extraordinary control points. An extraordinary control point and its corresponding extraordinary region are show in the figure. To evaluate the surface in extraordinary region, special treatment has to be applied [25]. In the very first definition of T-splines, a subdivision rule is applied to evaluate the surface in extraordinary region. This kind of model is called T-NURCCs (Nonuniform rational Catmull–Clark surfaces with T-junctions). And latter, as T-splines with extraordinary control points become commonly used, this kind of model is also called T-splines. As the subdivision rule used in T-NURCCs is not quite compatible with NURBS, several research have been reported about handling extraordinary control points by using template [26] or knot interval duplication [27] to avoid subdivision.

For the treatment of extraordinary control points, refer [27] for a good survey. And in this work, we handle the extraordinary control points with the original T-NURCCs subdivision rule. And thus the entire surface model is at least C^2 continues except the extraordinary position. The extraordinary position is the position on the surface associated with the extraordinary control point on the control mesh. At the extraordinary position, the surface is G^1 continues. For other extraordinary control points handling methods, the obtained continuity in extraordinary regions might be worse than the subdivision based method. But as long as the continuity is not worse than G^1, the machining error at the patch boundaries can be ensured by ensuring the error of each patch. Thus, the proposed method is still capable of generating a suitable tool path.

The proposed algorithm regards the input model as a few rectangular patches. A pair of adjacent patch shares an entire edge of the patch. This perspective is consistent with many modeling approaches [23, 24]. Notice that if there are too many extraordinary control points or holes in the mesh, the size of a patch might be too small. In this situation, the tool path generated by the proposed method will be less efficient as we utilize a patch-by-patch machining process. But, as the T-spline design usually tries to minimize the number of extraordinary control points, the proposed algorithm is suitable in general.

2.3. Evaluation of the Geometric Properties. Local geometric properties of the surface are useful in path generation. In order to evaluate local geometric properties of the surface, partial differential of the surface should be established. The derivation of the geometric properties of T-splines can be found in [19]. For equation (1), we use the notation $\mathbf{B} = \sum_{i=1}^{n} w_i B_i(u,v)\mathbf{P}_i$, $W = \sum_{i=1}^{n} w_i B_i(u,v)$ and $\mathbf{S} = \mathbf{B}/W$. The first and second derivatives of the T-spline surface are as follows:

$$\begin{cases} \mathbf{S}_u = \dfrac{\mathbf{B}_u - W_u\mathbf{S}}{W}, \\[2ex] \mathbf{S}_v = \dfrac{\mathbf{B}_v - W_v\mathbf{S}}{W}, \\[2ex] \mathbf{S}_{uu} = \dfrac{\mathbf{B}_{uu} - 2W_u\mathbf{S}_u - W_{uu}\mathbf{S}}{W}, \\[2ex] \mathbf{S}_{uv} = \dfrac{W\left(\mathbf{B}_{uv} - W_{uv}\mathbf{S} - W_u\mathbf{S}_v\right) + W_uW_v\mathbf{S} - \mathbf{B}_uW}{W^2}, \\[2ex] \mathbf{S}_{vv} = \dfrac{\mathbf{B}_{vv} - 2W_v\mathbf{S}_v - W_{vv}\mathbf{S}}{W}. \end{cases} \tag{2}$$

The surface normal can be calculated as

$$\mathbf{n} = \frac{\mathbf{S}_u \times \mathbf{S}_v}{\left|\mathbf{S}_u \times \mathbf{S}_v\right|}. \tag{3}$$

And, the first and second fundamental forms of the surface are

$$\begin{cases} E = \mathbf{S}_u^2, \\ F = \mathbf{S}_u \mathbf{S}_v, \\ G = \mathbf{S}_v^2, \\ L = \dfrac{1}{\sqrt{EG - F^2}} \langle \mathbf{S}_{uu}, \mathbf{S}_u, \mathbf{S}_v \rangle, \\ M = \dfrac{1}{\sqrt{EG - F^2}} \langle \mathbf{S}_{uv}, \mathbf{S}_u, \mathbf{S}_v \rangle, \\ N = \dfrac{1}{\sqrt{EG - F^2}} \langle \mathbf{S}_{vv}, \mathbf{S}_u, \mathbf{S}_v \rangle. \end{cases} \tag{4}$$

In the equations above, $\langle \cdot \rangle$ denotes mixed product and $|\cdot|$ denotes Euclidean norm. And, the symbols with subscripts denote its partial derivatives. As the partial derivative and first and second fundamental forms are calculated, the geometric properties such as local curvature can be easily obtained using the principle of differential geometry.

For the extraordinary region, the geometric properties are obtained using the same equations. But some conversion of control points and basic functions are needed according to extraordinary region handling strategy. For subdivision-based method, the control mesh has to be refined several times using the local refinement rule of T-NURCCs [17]. After local refinement, a part of the original extraordinary region will be evaluable and the local geometric properties can be evaluated using the aforementioned equations according to the refined control mesh. For the remaining part of the original extraordinary region, the local refinement rule can be applied iteratively, until it becomes evaluable.

3. Tool Path Generation

3.1. Algorithm Overview. The workflow of the proposed tool path generation algorithm is shown in Figure 3. The input of the algorithm is a T-spline surface model probably with several extraordinary control points. And, the output is a nonretraction tool path. The proposed algorithm includes the following steps. The connectivity of patches is analysed first. The topology layout of patches is obtained and represented as an undirected weighted graph. Then, the patch order is determined by solving a TSP problem using LKH solver. The transit corners of the tool path are selected using a back-tracking algorithm. For simple models, these steps can also be manually finished in consideration of the machining time or product texture. After patch order and transit corners are determined, the tool path is generated patch by patch using isoparametric strategy. In this step, we proposed a scallop tuning algorithm to make the tool path finish at the selected transit corner. In this work, we consider using ball-end mill as the tool. And, the posture of the tool is set to avoid gouge.

3.2. Patch Neighboring Relation Analysis. In the T-spline surface model, there are two kinds of patch neighboring relation, corner connection, and boundary connection. Two patches have a boundary connection means they share a boundary edge. And, two patches have a corner connection means they share a corner node but do not have a common boundary edge. The connection of the patches can be represented as an undirected weighted graph. An example of a patch layout and the corresponding undirected weighted graph is shown in Figure 4. And, patches are colored according to its neighboring relation with patch 0. The blue patches are boundary connected with patch 0. The green patches are corner connected with patch 0. And, gray patches are not connected with patch 0. In the undirected weighted graph, two kinds of connections can be distinguished using different weights. And, they are represented using different kinds of lines in the figure.

Except for the neighboring relation, we also care about which corner or boundary edge of a patch is shared with the other. To denote corners and edges of a patch, consider a parametric surface patch whose parameter domain is a unit square in R^2 $([0,1] \times [0,1])$; the corner with coordinate $u = 0, v = 0$ is called the lower left corner of this patch. Similarly, the boundary edge with the algebraic expression $u = 1$ is called the right boundary edge of this patch. For a patch neighboring relation, it has to be clear which corner or boundary is shared by the two patches. This information usually can be obtained from the T-splines' data structure. And, in the following discussion, we consider it as obtainable for algorithms.

3.3. Patch Order Determination. To generate a nonretraction tool path, the patch processing order has to be determined. The output of patch order determination and transit corner selection is a processing list containing several processing data element. And, each processing data element in the list contains the information including processing patch (patch ID), starting corner, finishing corner, and preferred crossfeed direction. The patch order and preferred crossfeed direction are first determined, and the transit corners are selected later. One limiting condition here is that the starting corner of a processing data element must be the same point as the finishing corner of the former processing data element. The preferred crossfeed direction in the record is used to define which crossfeed direction will be utilized when both crossfeed directions are acceptable. Consider that this process might be manually designed or modified especially when the patch layout is simple. For implementation, the processing list is saved as a file and input to the subsequent isoparametric tool path generation procedure.

The patch order has to ensure that two patches that are processed successively are adjacent in position. This problem can be regarded as a traveling salesman problem (TSP) to some degree. The TSP problem can be described as given a list of cities and the distance between each pair of cities and find the shortest possible route that visits each city exactly once and returns to the origin city. Here, we regard each patch as a city and regard each neighboring relation as a road

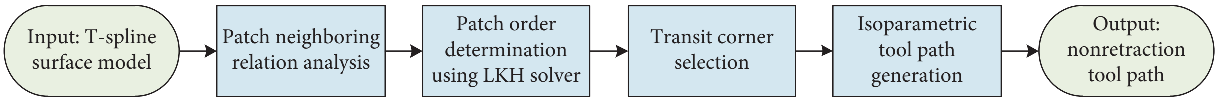

Input: T-spline surface model
Patch neighboring relation analysis
Patch order determination using LKH solver
Transit corner selection
Isoparametric tool path generation
Output: nonretraction tool path

Figure 3: The workflow of the proposed tool path generation algorithm.

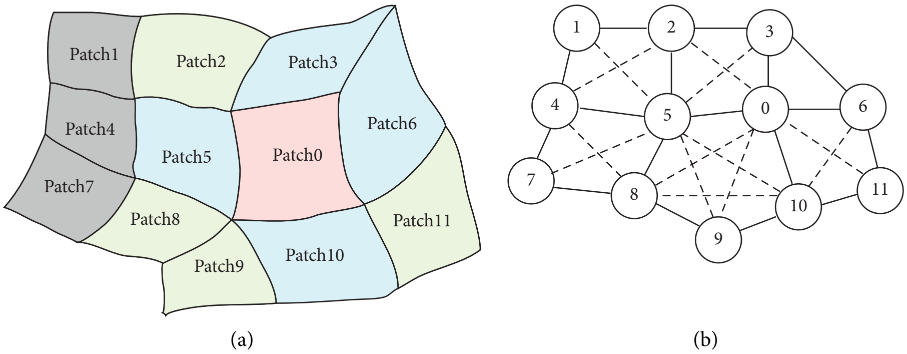


Figure 4: The neighboring relation of a model with 12 patches; (a) the patch layout, for each patch, its neighboring relation with patch 0 is represented in the figure using a different color; (b) the undirected weighted graph generated from the neighboring relation in (a); boundary connections are represented using solid lines and corner connections are represented using dashed lines.

between cities. And, convert the patch order determination problem into a TSP problem. The neighboring relation between patches can be obtained from the undirected weighted graph. The distance weight of the same kind of connection is set to be the same value. To facilitate transit corner selection, boundary connection is preferred than corner connection. Thus, a large distance weight is assigned to a corner connection edge. And, because the tool path does not need to go back to the original position like the TSP problem, we add a zero-weighted link between the starting patch and the suggested finishing patch. The starting patch and suggested finishing patch can be set manually or randomly decided. The weight of this link is set to be zero because we prefer it to appear at the end of the TSP solution to virtually link the finishing patch and the starting patch. If this link appears at the beginning of the TSP solution, the patch order in the solution (except the first one) has to be reversed to put the finishing patch at the end.

To solve the TSP problem here, we use the LKH solver. The LKH is an efficient TSP solver which is published online. With LKH, for a typical 100-city problem, the optimal solution can be found in less than a second. It is also used to generate a kind of TSP tool path for freeform surfaces in other research [28, 29]. One possible zero-weighted link and a corresponding TSP solution for the graph in Figure 4 is shown in Figures 5(a) and 5(b). Figures 5(c) and 5(d) give an patch layout that a corner connection must be included in a TSP solution.

The preferred crossfeed direction is used to facilitate users to choose the crossfeed direction to optimize the machining time or control the machining texture. As default, the preferred crossfeed direction can be set to be u-direction or the direction results in a shorter isoparametric tool path to process this patch.

3.4. Transit Corner Selection. After the patch order is determined, the starting corner and the finishing corner have to be defined for each patch. These starting and finishing corners are the transit corners of the entire tool path to connect the adjacent patches. The proposed transit corner selection algorithm roughly consists of two parts, main loop of the algorithm and the back tracking part. In the main loop, the starting corner of the first patch (the starting corner point of the entire tool path) is randomly defined. Then, if the current patch and the next patch share a boundary edge, there are two possible situations and there will be at least one possible finishing corner for the current patch. If the current patch and the next patch only share a corner, there are three different situations. All these five situations are shown in Figure 6. The finishing corner will be selected from all the possible finishing corners. If there are two possible finishing corners, the one which is compatible with the preferred crossfeed direction will be selected. If both of them are compatible with the preferred crossfeed direction, the finishing corner will be randomly selected. After the finishing corner of the current patch is determined, the starting corner of the next patch is determined and the finishing corner of the next patch can be selected in the same way. This process continues until the starting corner of the last patch is determined. And, the finishing corner of the last patch can be randomly selected. For a patch connected with the next patch with a corner connection, if the shared corner is the same as its starting corner, there is no possible finishing corner. In this situation, the algorithm turns into the back tracking loop and tries to back track to the former patch, set the currently used finishing corner as unusable, and keep searching for the possible solution. If the algorithm goes back to the first patch and still could not find a possible solution, the current patch order cannot generate a

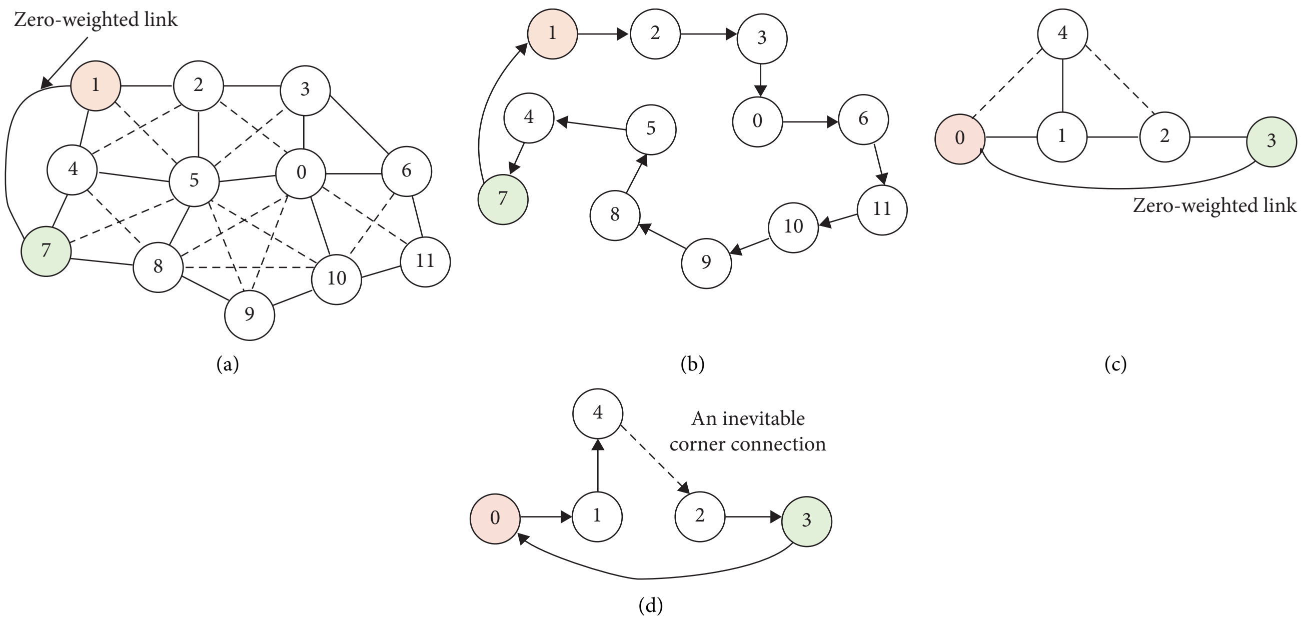


Figure 5: The patch order obtained by solving TSP problem, the starting patch is marked in red, the suggested finishing patch is marked in green, and a zero-weighted link is added; (a) the patch layout in Figure 4; (b) one possible TSP solution for the graph in (a); (c) a patch layout that corner connection link must be included in the TSP solution; (d) one possible TSP solution for the graph in (c).

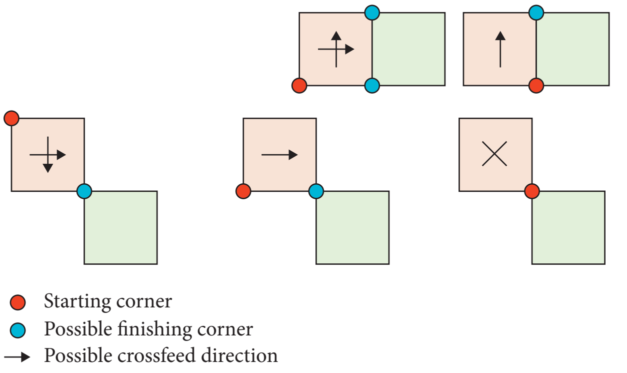


Figure 6: Five possible situations for determining the finishing corner. The current patch are drawn in red and the next patch is drawn in green.

nonretraction tool path. In this situation, it is suggested to adopt a retraction at the patch where the path generation is failed or tries to select another starting patch and generate a new patch order. Fortunately, this is not likely to happen, as we always take the boundary connection as the first choice in TSP solution. The flowchart of the transit corner selection algorithm is shown in Figure 7. And, one possible result of the patch layout in Figure 4 with the patch order in Figure 5 is shown in Figure 8.

3.5. Isoparametric Tool Path Generation. After the processing order is determined, the remaining task is to generate a tool path with the defined starting corner, finishing corner, and preferred crossfeed direction for each patch. This can be achieved easily using the isoparametric strategy. If the starting and finishing corners are at the same side of the patch, there is only one possible crossfeed direction. And, if the starting and finishing corners are located in the diagonal of the patch, the preferred crossfeed direction in the processing data record will be selected as the crossfeed direction.

Given the starting corner, the crossfeed direction, the tool radius, and the acceptable scallop height, there are various methods to generate an isoparametric tool path for a parametric surface with rectangular topology [19, 28]. Consider the situation that the lower left corner is the starting corner and the crossfeed direction is the u-direction. The first tool path has the algebraic expression $u = 0$. Then, a side step Δu in parameter space is calculated according to the tool radius, the scallop height, and the geometric property of the surface. To calculate the side step Δu, we first sample a number of points on the current tool path. Then, for each point, calculate the side step in physical space using the following equation:

$$g = \sqrt{\frac{8hrR}{R + \mathrm{sign} \cdot r}}. \tag{5}$$

As shown in Figure 9, g is the side step in physical space, r is tool end radius, R is the surface curvature, h is the scallop height, and sign is an operator whose value is 0 for flat surface, 1 for convex surface and −1 for concave surface. And, Δu can be calculated for each sampling point by mapping the side step g in physical space back into the parameter space. The calculation formula of Δu is given in equation (6). And, for patches with v-direction as the crossfeed direction, the calculation formula of Δu is also given in equation (6). The minimum Δu among them is utilized as the side step in parameter space, and the second tool path can be generated by offset the first tool path with a

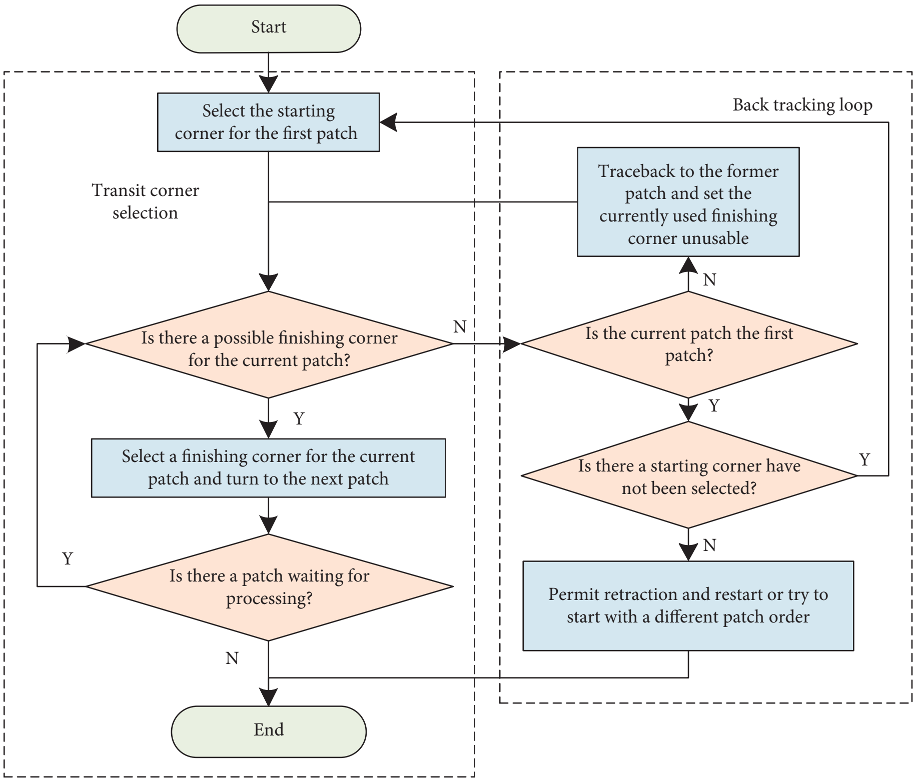

Start
Select the starting corner for the first patch
Back tracking loop
Transit corner selection
Traceback to the former patch and set the currently used finishing corner unusable
N
Is there a possible finishing corner for the current patch?
N
Is the current patch the first patch?
Y
Y
Select a finishing corner for the current patch and turn to the next patch
Is there a starting corner have not been selected?
Y
Y
Is there a patch waiting for processing?
N
Permit retraction and restart or try to start with a different patch order
N
End

Figure 7: The flow chart of the transit corner selection algorithm.

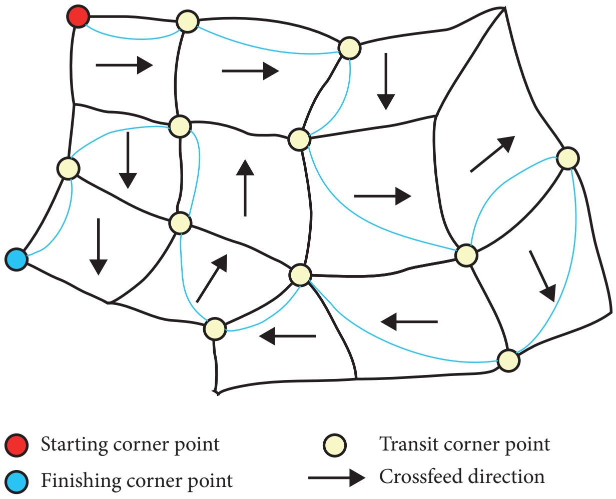
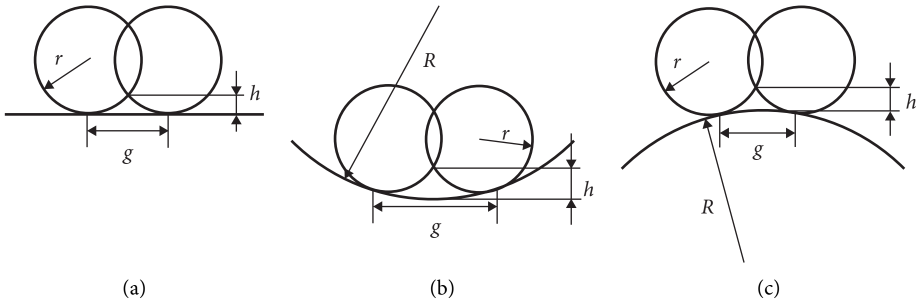


Figure 8: One possible processing order generated using the proposed algorithm for the patch layout in Figure 4 and the patch order in Figure 5. The blue curves in the figure connect the starting and finishing corners for each patch.

parametric distance Δu. And a series of iso-parametric lines can be generated by repeatedly performing this process. These isoparametric lines are the tool paths of this surface patch:

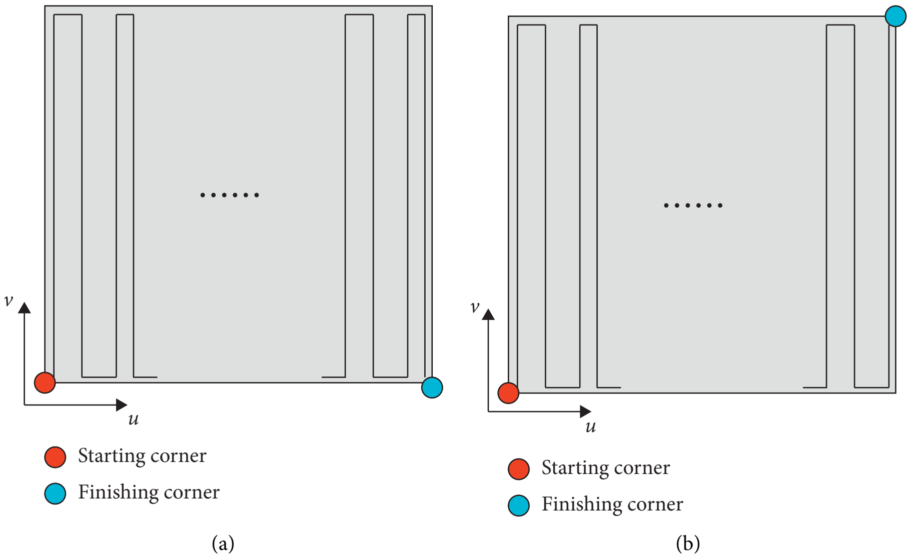
$$\begin{cases} \Delta u = \dfrac{gF}{\sqrt{E^2G - EF^2}}, \\ \\ \Delta v = \dfrac{-gF}{\sqrt{EG^2 - F^2G}}. \end{cases} \qquad (6)$$

To avoid retraction, the isoparametric lines are usually connected in a zig-zag way. As shown in Figure 10, for an isoparametric tool path starting from its lower left corner with u-direction as the crossfeed direction, there are two possible finishing corners. Before path generation, we cannot predict which corner will the zig-zag path finish at. In order to ensure the tool path finished at the expected corner, we propose utilizing a scallop tuning for patches that the isoparametric tool path does not finish at the expected finishing corner. Obviously, the finishing corner is decided by the odevity of



(a) (b) (c)

Figure 9: Side step calculation under scallop limit.



(a) (b)

Figure 10: Two different finishing corners of the isoparametric tool path starting from the lower left corner with u-direction as the crossfeed direction.

the number of the isoparametric lines. If the path number of the generated isoparametric lines is n and the finishing corner is not the expected one, generate a tool path with $n+1$ isoparametric lines correcting the finishing corner. Thus, we slightly decrease the scallop height used in the tool path generation of this patch. For instance, if the original scallop is h, it is first decreased by a constant step length such as $0.1h$. Then, an isoparametric tool path is generated with $0.9h$ scallop. If the number of isoparametric lines is still n, the scallop will be decreased again with the step length 321. If the number of isoparametric lines is $n+1$, the generated tool path of 322 can be used as the final result. Otherwise, if the number of isoparametric lines is greater than $n+1$, the bisection method can be utilized to find a proper scallop between the current scallop and the original scallop and produce a tool path with $n+1$ isoparametric lines.

For five-axis machining, the tool posture has to be determined for a tool path. As we consider ball-end mill as the machining tool in this work, the posture is set to avoid gouge. Without loss of generality, the direction of the tool axis direction which is gouge free and most close to the Z-axis is set to be the tool axis direction for five-axis machining. For other kinds of tools, posture can also be determined by other standard such as surface curvature matching.

4. Experimental Results

To verify the effectiveness of the proposed algorithm, experimental results are given in this section. The test model is a keycap model with five T-spline patches and four extraordinary control points. As the work focuses on the machining of freeform surfaces, the model ignores the connecting structure of the keycap and only uses the shell part. The procedure of experiment is shown in Figure 11. Different patches in the model are shown in different color. The model is first input to the test program. The program generates the five-axis tool path using the proposed algorithm, and the corresponding Gcode file is also generated. The Gcode file is first simulated in Varicut software. And, the molds are machined using a five-axis machining center. After the molds are manufactured, the injection experiment is conducted on an injection molding machine to get the final product.

The machining center used here is a Mikron UCP 600 five-axis machining center. The injection molding machine

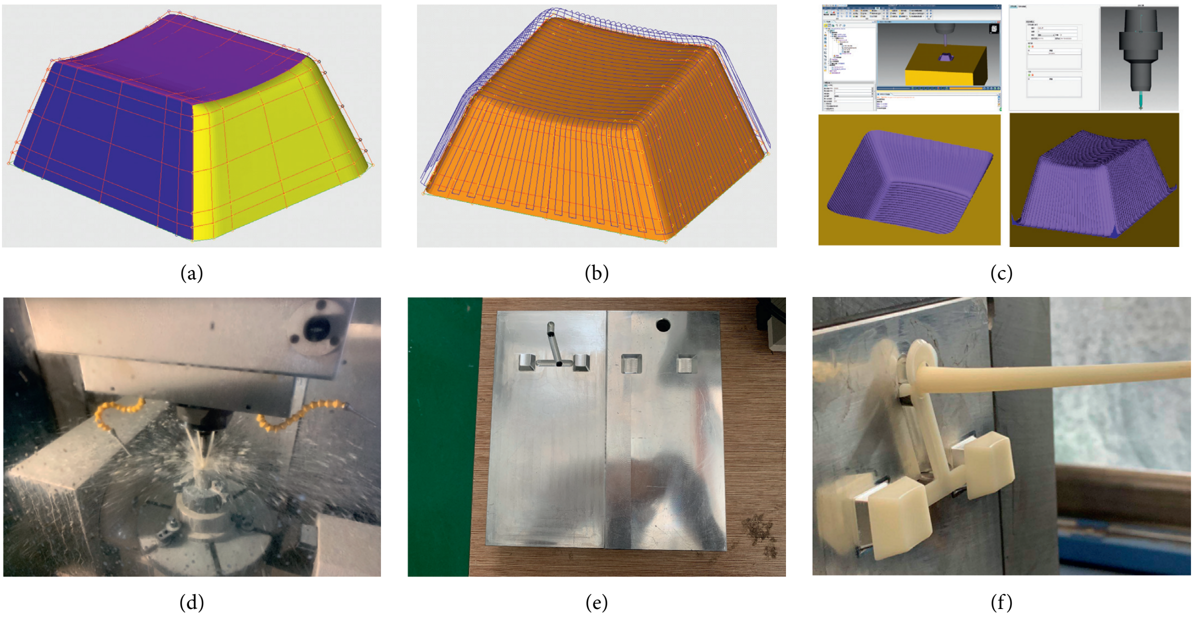


Figure 11: The procedure of experiment. (a) Input T-spline model, (b) tool path generation, (c) tool path simulation, (d) five-axis machining, (e) injection molds, and (f) polymetric product.

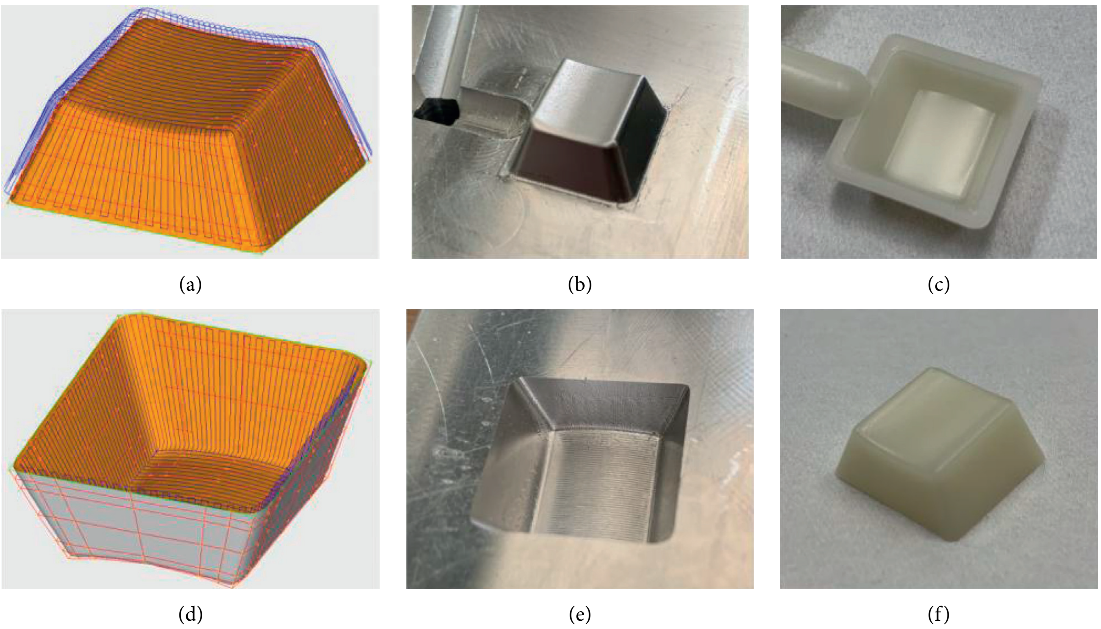


Figure 12: Tool path, injection molds, and final ABS product for the keycap model.

is ONGO Z70/JD. Mold material is 6061 aluminium alloy. The injection molding material is ABS. The convex model is in size of 14.4 mm × 14.4 mm × 6.8 mm, and the concave model is in size of 18 mm × 18 mm × 8.6 mm. To avoid local gouge or overcut, the finish machining cutting tool is a ball-end mill with 0.5 mm end radius. And, the scallop height limit is set to be 0.005 mm.

The experimental results are shown in Figure 12. The proposed method successfully generates a nonretraction tool path for the T-spline surface model with extraordinary control points. In each surface patch, the tool path is a zig-zag isoparametric tool path. These tool paths are end to end. And, the machining of the molds is finished at once without retraction. As shown in Figure 11, the boundaries of the T-spline patches are located near the edges of the keycap model. It can be seen in Figure 12 that the quality of the molds and the polymeric products is quite good in all places. And, there is no visible defect at these patch boundaries.

5. Conclusions

In this paper, we proposed a new five-axis tool path generation algorithm facing the T-spline surface model with extraordinary control points. Mold manufacturing is a typical application of this algorithm. One major advantage of using T-splines instead of NURBS is that the mold quality can be well controlled especially at patch boundaries as the T-spline surface model has high level continuity all over the model. Besides, the proposed algorithm utilizes the patch layout information of T-spline models to generate a nonretraction tool path. And, we also performed the mold machining and injection test to manifest the proposed method.

So far, there are still lots of potential improvements to the current work. Firstly, the proposed method still generates the tool path in a patch-by-patch manner. One possible improvement is to merge the tool path in adjacent patch to further reduce the processing time. Machining texture is also a worthwhile topic to improve the surface quality of the model. In the experiment, we can see some machining texture which will somehow influence the surface smoothness of the final product. Another interesting research direction is the T-spline-based mold design. However, there are already many T-spline modeling algorithms and T-spline-based CAD software such as Rhino and Fusion 360. They are not facing the injection mold design situation. Functions such as parting surface design are needed for mold design.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the National Nature Science Foundation of China (no. 51575483), Science Fund for Creative Research Groups of National Natural Science Foundation of China (no. 51821093), Key Research and Development Plan of Zhejiang Province (no 2018C01073), and the Fundamental Research Funds for the Central Universities (no. 2019QNA4004).

References

[1] C. Fernandes, A. J. Pontes, J. C. Viana, and A. Gaspar-Cunha, "Modeling and optimization of the injection-molding process: a review," *Advances in Polymer Technology*, vol. 37, no. 2, pp. 429–449, 2018.

[2] J. Gómez-Monterde, M. Sánchez-Soto, and M. L. Maspoch, "Influence of injection molding parameters on the morphology, mechanical and surface properties of ABS foams," *Advances in Polymer Technology*, vol. 37, no. 8, pp. 2707–2720, 2018.

[3] F. Liu, Y. Zhang, Y. Zhang et al., "Experimental investigation of the effects of process conditions on the morphology in injection molding of polymer blends," *Advances in Polymer Technology*, vol. 36, no. 3, pp. 341–351, 2017.

[4] M. C. Song, Z. Liu, M. J. Wang, T. M. Yu, and D. Y. Zhao, "Research on effects of injection process parameters on the molding process for ultra-thin wall plastic parts," *Journal of Materials Processing Technology*, vol. 187-188, pp. 668–671, 2007.

[5] L. Guangming, F. Hui, Z. Lixuan, and Y. Bin, "Research on optimal design of the injection mold parting direction based on preference relation," *The International Journal of Advanced Manufacturing Technology*, vol. 79, no. 5–8, pp. 1027–1034, 2015.

[6] I. Matin, M. Hadzistevic, J. Hodolic, D. Vukelic, and D. Lukic, "A CAD/CAE-integrated injection mold design system for plastic products," *The International Journal of Advanced Manufacturing Technology*, vol. 63, no. 5–8, pp. 595–607, 2012.

[7] Z. Li, X. Zhou, W. Liu, Q. Niu, and C. Kong, "A similarity-based reuse system for injection mold design in automotive interior industry," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 5-8, pp. 1783–1795, 2016.

[8] X. Pessoles and C. Tournier, "Automatic polishing process of plastic injection molds on a 5-axis milling center," *Journal of Materials Processing Technology*, vol. 209, no. 7, pp. 3665–3673, 2009.

[9] Y. Jiang, L. Zhenliang, and M. Li, "Application of fuzzy and rough sets theory in the optimization of machining parameters for mold milling operations," *The International Journal of Advanced Manufacturing Technology*, vol. 28, pp. 1071–1077, 2006.

[10] H. U. Cho, J. W. Park, and Y. C. Chung, "Computation of optimal tool length for 5-axis ball-ended milling of molding die," *Journal of Mechanical Science and Technology*, vol. 26, no. 10, pp. 3097–3101, 2012.

[11] L. A. Sarmiento-Merida, A. Guevara-Morales, and U. Figueroa-López, "Determining the optimum parting direction in plastic injection molds based on minimizing rough machining time during mold manufacturing," *Advances in Polymer Technology*, vol. 37, no. 1, pp. 194–201, 2018.

[12] A. Lasemi, D. Xue, and P. Gu, "Recent development in CNC machining of freeform surfaces: a state-of-the-art review," *Computer-Aided Design*, vol. 42, no. 7, pp. 641–654, 2010.

[13] W. He, M. Lei, and H. Bin, "Iso-parametric CNC tool path optimization based on adaptive grid generation," *The International Journal of Advanced Manufacturing Technology*, vol. 41, no. 5-6, pp. 538–548, 2009.

[14] S. Sarkar and P. P. Dey, "A new iso-parametric machining algorithm for free-form surface," *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 228, no. 3, pp. 197–209, 2014.

[15] S. Ding, M. A. Mannan, A. N. Poo, D. C. H. Yang, and Z. Han, "Adaptive iso-planar tool path generation for machining of free-form surfaces," *Computer-Aided Design*, vol. 35, no. 2, pp. 141–153, 2003.

[16] Z. Lin, J. Fu, H. Shen, and W. Gan, "A generic uniform scallop tool path generation method for five-axis machining of freeform surface," *Computer-Aided Design*, vol. 56, pp. 120–132, 2014.

[17] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, "T-splines and T-NURCCs," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 477–484, 2003.

[18] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson, "Watertight trimmed NURBS," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–8, 2008.

[19] W.-F. Gan, J.-Z. Fu, H.-Y. Shen, Z.-Y. Chen, and Z.-W. Lin, "Five-axis tool path generation in CNC machining of -spline surfaces," *Computer-Aided Design*, vol. 52, pp. 51–63, 2014.

[20] G. Zhao, Y. Liu, W. Xiao, O. Zavalnyi, and L. Zheng, "STEP-compliant CNC with T-spline enabled toolpath generation capability," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 5–8, pp. 1799–1810, 2018.

[21] G. Zhao, O. Zavalnyi, Y. Liu, and W. Xiao, "Prospects for using T-splines for the development of the STEP-NC-based manufacturing of freeform surfaces," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 1–4, pp. 1–16, 2019.

[22] J. Feng, J. Fu, Z. Lin, C. Shang, and B. Li, "Direct slicing of T-spline surfaces for additive manufacturing," *Rapid Prototyping Journal*, vol. 24, pp. 709–721, 2018.

[23] H. Wang, Y. He, X. Li, X. Gu, and H. Qin, "Polycube splines," *Computer-Aided Design*, vol. 40, no. 6, pp. 721–733, 2008.

[24] M. Campen and L. Kobbelt, "Quad layout embedding via aligned parameterization," *Computer Graphics Forum*, vol. 33, no. 8, pp. 69–81, 2014.

[25] X. Li, F. Chen, H. Kang, and J. Deng, "A survey on the local refinable splines," *Science China Mathematics*, vol. 59, no. 4, pp. 617–644, 2016.

[26] W. Wang, Y. Zhang, G. Xu, and T. J. R. Hughes, "Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline," *Computational Mechanics*, vol. 50, no. 1, pp. 65–84, 2012.

[27] L. Liu, Y. J. Zhang, and X. Wei, "Handling extraordinary nodes with weighted T-spline basis functions," *Procedia Engineering*, vol. 124, pp. 161–173, 2015.

[28] Z. Lin, J. Fu, H. Shen, W. Gan, and S. Yue, "Tool path generation for multi-axis freeform surface finishing with the LKH TSP solver," *Computer-Aided Design*, vol. 69, pp. 51–61, 2015.

[29] Z. Lin, J. Fu, Y. Sun, Q. Gao, G. Xu, and Z. Wang, "Non-retraction toolpath generation for irregular compound free-form surfaces with the LKH TSP solver," *The International Journal of Advanced Manufacturing Technology*, vol. 92, no. 5–8, pp. 2325–2339, 2017.