The International Journal of Advanced Manufacturing Technology (2019) 104:599–615
https://doi.org/10.1007/s00170-019-03923-8

ORIGINAL ARTICLE

A new approach for uncut detection and automatic design of EDM electrodes

Junjie Li[1] · Xiong Hui Zhou[1] · Wei Liu[1] · Hengyuan Ma[1]

Received: 8 October 2018 / Revised: 29 April 2019 / Accepted: 21 May 2019 / Published online: 6 June 2019

Abstract

Plastic parts such as smart cellphones are becoming more compact and variable than before. The structures of injection molds, that produce those plastic parts, also become more compact and complex correspondingly. With the restriction of the cutter radius and feeding direction, the conventional milling process usually fails to process such variable structures of injection molds, and the electric discharge machining (EDM) electrodes are always employed in these cases. Although specific commercial CAD/CAM software for injection mold application is widely used in electrode design, large amounts of human–computer interactions are still needed. Thus, there is still room for improvement in the EDM electrode design. This paper presents a systematic methodology which includes uncut detection in the injection mold and automation of EDM electrode design. A toolpath simulation method is proposed to check all the faces of an injection mold to find the uncut regions that cannot be processed by all the cutters. Then, each uncut region is recursively split into sub-regions to construct an electrode with a set of CAD modeling operations. Based on diverse shapes of the sub-regions, two geometry operation methods, extruding method and cutting method, are employed to generate electrode tips. Case studies and industrial applications have demonstrated the algorithm's powerful ability in handling complex structures of injection mold, and the efficiency for EDM electrodes design could be improved by 70% at least.

Keywords Die and mold CAD · Uncut region detection · Toolpath simulation method · EDM electrode design · Design automation

1 Introduction

Injection molding parts have been widely used in aerospace, automobile, and consumer products due to the lower cost [1]. On the other hand, those parts are becoming more compact in size and intricate in shape. The injection molds, which produce these plastic parts, are also becoming more variable correspondingly. With the restriction of the cutter radius and feeding direction, the conventional milling process is not good at machining such delicate regions containing sharp corners [2, 3], hereinafter referred to as "uncut region." Therefore, EDM is now more frequently used in the injection mold manufacturing process because of its power ability in processing such variable structures [4, 5].

However, EDM needs extra design and machining of electrodes, which is a delicate, time-consuming, and error-prone work. For example, over 100 sets of electrodes are required to manufacture a single mold of a mobile phone cover [6]. Actually, the cost and time consumption in the mold machining by EDM are mainly the design and manufacture of electrodes, accounting for over 50% of the total machining cost [7].

Usually, the EDM engineers find the uncut regions manually and design electrodes with a lot of human–computer interactions. The experiences of EDM engineers play an important role in the processing of uncut region detection and electrode design. Therefore, for the above typical product, roughly 80 h are spent on the electrode design process [8], which is becoming the bottleneck in improving production efficiency and reducing lead time. Currently, in spite of the widespread use of the CAD/CAM software and the customized software for electrode design in the mold design and manufacture [9], due to the delicate and complex structures of injection molds, the electrodes are generated manually that could result in bad results. For instance, many uncut regions are ignored without generating electrodes, or the created electrode is difficult to be machined.

This paper proposes a new approach for uncut detection and automatic design of EDM electrodes, in which a toolpath simulation method that supports the automation of uncut

✉ Xiong Hui Zhou
xhzhou@sjtu.edu.cn

[1] National Engineering Research Center of Die & Mold CAD, Shanghai Jiao Tong University, Shanghai 200030, China

detection, and two computer-aided geometric modeling methods for generating electrode tips are addressed.

2 Previous work

Various aspects of the automation of the design and manufacturing of plastic injection molds have been studied. The determination of the parting direction and the design of the parting surface and parting line are the major tasks in the early stages of the mold design process. Quang and Lin [10] proposed a new slicing approach that uses multiple sets of cutting planes to automatically determine parting lines. The orbital motion of the electrode makes the cavity dimension in the workpiece deviate from the design objective. Recently, the electrode compensation techniques are studied to solve the problem. Wang et al. [11] presented the non-uniform offset method to modify the geometric shape of a free-form electrode to eliminate the effects of electrode orbiting. Not only did Sriani et al. [12] adjust the electrode for machining overcut but also the electrode has been considered for the chosen orbit motion into geometry design. Lo and Jiang [13] proposed an electrode compensation technique according to the envelope theory to eliminate deviation errors. In addition, the integration of CAD/CAPP/CAM is the development trend for electrodes. Ma et al. [9] presented a feature-based integration approach of CAD/CAPP/CAM, and a system with multi-layer features has been realized.

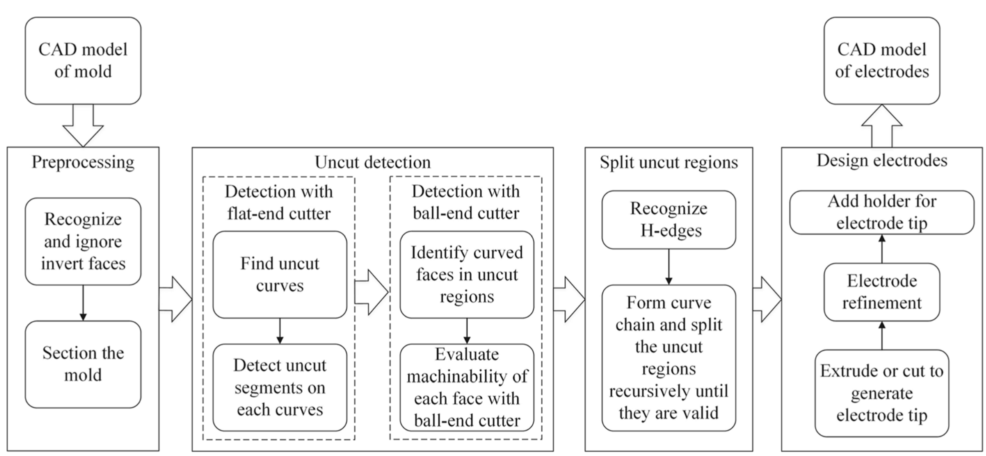
Though the EDM process has been studied extensively, very limited work has been carried out on the geometric design of EDM electrodes or the automation of the design process. Ding et al. [14, 15] proposed an approach to detect the sharp corner on mold which should be processed with EDM, and automate the design of electrodes. But the method cannot detect all the uncut regions and also failed to deal with the problem of leftovers after EDM (as addressed in Sect. 5). Mahajan et al. [16] investigated the principles of designing a knowledge-based system for automated EDM electrode design. Lee and Li [8] developed a geometric algorithm that can split a single electrode into multiple electrodes automatically to avoid machinability problems, in which the problem of leftovers was also taken into account for the first time. However, this method cannot overcome the disadvantage of most hint-based approaches; that is, it is not efficient enough for complex parts.

Zhou and Zhang [17] proposed a volume decomposed-based algorithm to automate the electrode splitting and generate electrode CAD models. But the algorithm assumed that the uncut regions have been detected in advance. Geng and Liu [18] proposed an approach to extract the machining feature of the region containing internal sharp points and to construct electrodes based on the features. But this method did not take the uncut region's size into consideration; therefore, the generated electrodes sometimes are too large to be machined (as addressed in Sect. 4.3). In the interference detection, Lee et al. [19] detected the interference regions by means of curve offset. Ding et al. [20] proposed a method to detect local interference and global interference in 3-axis mold machining. Review of related literature reveals that there is still room for improvement in the process of uncut detection and electrode design.

3 Methodology of uncut detection and electrode design

Usually, a mold is first machined by the 3-axis milling center, and then the area that cannot be machined by any cutter will be processed by EDM. In other words, EDM is used to process the regions which cannot be machined by the milling cutter. Based on this fact, a toolpath simulation method is proposed to simulate the tool machining process and detect the uncut regions. The prerequisite for the method is that the invert surfaces of a mold are ignored (as addressed in Sect. 4). To detect the uncut regions, a mold is first sectioned with a set of paralleled planes. For any layer of section curves, the uncut segments will be detected. Therefore, the problem of uncut

Fig. 1 Methodology of uncut detection and electrode design



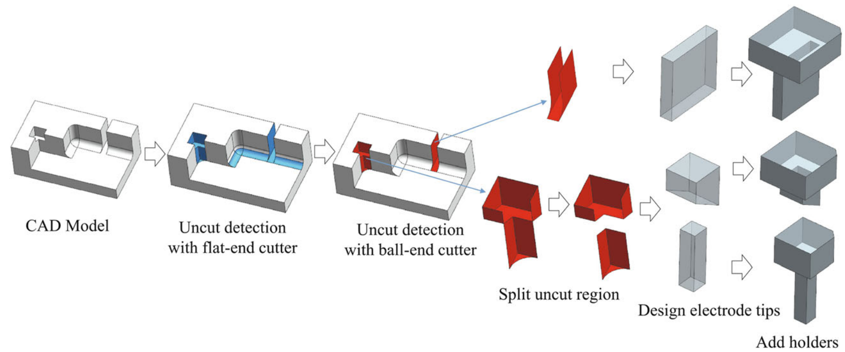


Fig. 2 Illustration of the methodology

detection on a 3D model is innovatively converted into a 2D problem, thus reducing the enormous computational burden.

As the cutter's type and diameter differ, the uncut regions change accordingly. The intersections of all the uncut regions with different cutters can be recognized as the uncut regions of the mold. In fact, the uncut detection with a flat-end cutter is more effective than detection with a ball-end cutter. Therefore, the flat-end cutter is used to detect uncut regions first. The uncut regions, which cannot be machined by the flat-end cutter, are detected with the ball-end cutter.

After uncut detection, all the uncut regions are isolated on the surfaces of the injection mold. Usually, one electrode can be generated for one uncut region. However, in most cases, the electrode itself may also be difficult to be machined for the existence of sharp corners or concave edges if three-axis machining is used. If the electrode generated for an uncut region is machinable, then this uncut region is called a *valid region*, otherwise an *invalid region*. Thus, the region containing convex edges is invalid, otherwise valid. The convex edges in the uncut region are called H-edges [8]. Lee and Li proposed the H-edge-based algorithm to split the invalid uncut region into several sub-regions recursively until all the uncut sub-regions are valid. In the H-edge-based algorithm, the more H-edges there are, the more time-consuming the splitting process is. In Lee and Li's opinion, a surface on a mold can only be marked as a machinable surface, or uncut, surface. The surface, which contains both uncut region and machinable region, is also marked as uncut. That makes the huge increase in the number of uncut regions and H-edges. This paper improves the H-edge-based algorithm proposed by Lee and Li. In the toolpath simulation method, the uncut regions are detected more precisely. A surface can be divided into the uncut region and machinable region, which can reduce the uncut area and H-edges, thus simplifying the splitting process.

A qualified electrode must be machinable, and the discharging area on it must be bigger than the corresponding uncut region in order to avoid the problem of leftovers (as addressed in Sect. 5). Two geometry operation methods for generating qualified electrode tips, extruding the uncut region or using the uncut region to cut the bounding box, are proposed based on the shape of each uncut region. The two methods are adaptable for all kinds of uncut regions; furthermore, the designing process is rapid and automated, without human–computer interactions.

Before detection, the invert faces are recognized and excluded in the following steps. There is still no need to design electrodes to process those faces, and the invert faces are usually machined after rotation of mold or with the help of a sliding block. Then, the mold without invert faces are sectioned with a set of paralleled planes, and a set of layers of section curves are generated. Thus, the 3D uncut area detection problem is converted into 2D uncut segment detection, which reduces the difficulty of detection. The next step will detect the uncut segments on each layer of section curves with the flat-end cutter and ball-end cutter, which simulates the process of cutter machining. All the 2D uncut segments can

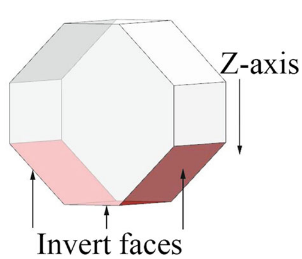


Fig. 3 The invert surfaces of mold

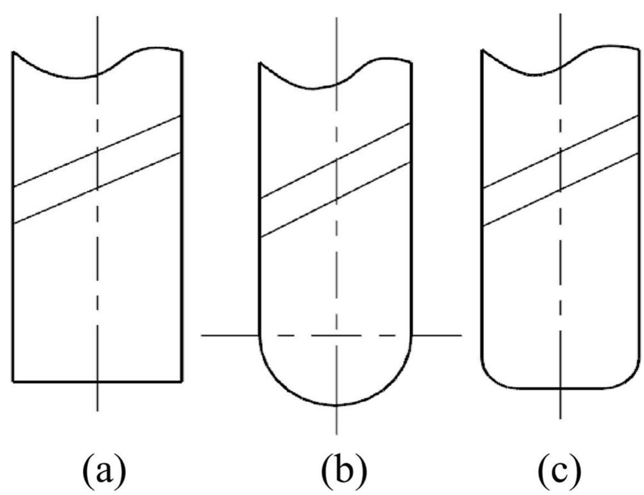


Fig. 4 **a** Flat-end cutter, **b** ball-end cutter, and **c** nose-end cutter

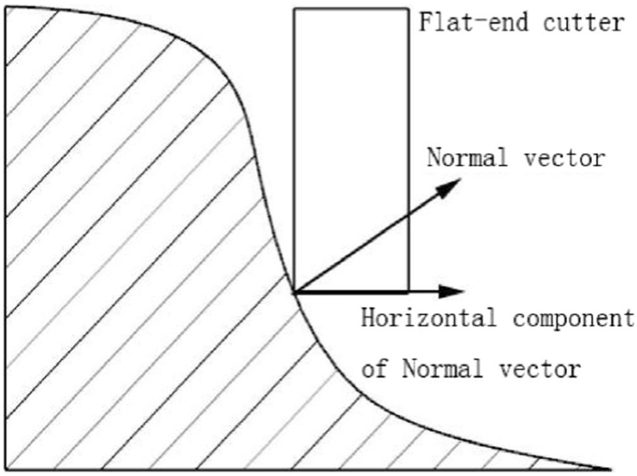


Fig. 5 Working process of the flat-end cutter

be used to represent the uncut regions of the mold. The uncut regions detected with this method are smaller than the uncut area detected on Lee and Li's method; on the other hand, the count of H-edges in the uncut regions can be reduced. Then, the invalid uncut regions will be split recursively until all the sub-regions are valid. Lastly, each electrode will be generated for each valid sub-region. The proposed methodology for uncut detection and automation of electrode design is illustrated in Fig. 1, and the result of each step is shown in Fig. 2. The input is the CAD model of a mold and three-axis milling cutters, then qualified electrodes will be generated as the output. Further details and key issues about these steps are discussed in the following sections.

4 Uncut detection for electrode design

This section focuses on the problem of how to detect uncut regions automatically. For the sake of explanation, it is assumed without loss of generality that the machining direction of the EDM process is along the z direction. If the angle between the normal vector of any point on the surface and z direction is less than 90°, this surface is called an *invert surface* as shown in Fig. 3, which cannot be machined by a 3-axis milling machine directly. Therefore, all the invert surfaces should be ignored before uncut detection. In fact, those invert surfaces usually are processed after rotation of the mold or with the help of a sliding block.

Due to the difference in shape and use, cutters can be categorized into three types: ball-end cutter, nose-end cutter, and flat-end cutter as shown in Fig. 4. If a profile cannot be machined by either flat-end or ball-end cutter, it cannot be machined by any other kinds of cutters [6]. Therefore, only the flat-end cutter and ball-end cutter are used to detect uncut regions in this paper.

4.1 Uncut detection with the flat-end cutter by the toolpath simulation method

When a point is processed by the flat-end cutter, the center of the cutter's bottom is located on the horizontal component of the normal vector at this point as shown in Fig. 5.

Theorem 1 *For a mold without invert surfaces, the necessary and sufficient condition for the existence of the uncut region of the flat-end cutter is that the bottom edge of the flat-end cutter intersects with the mold.*

Proof Consider the case where the uncut regions exist. For the processed point A as shown in Fig. 6 a, the flat-end cutter can intersect with the mold. For any point P in the interference body, the point Q is the projection of P to the bottom face of the flat-end cutter as shown in Fig. 6 a, and the point Q′ is the projection of P′ to the bottom face of the flat-end cutter as

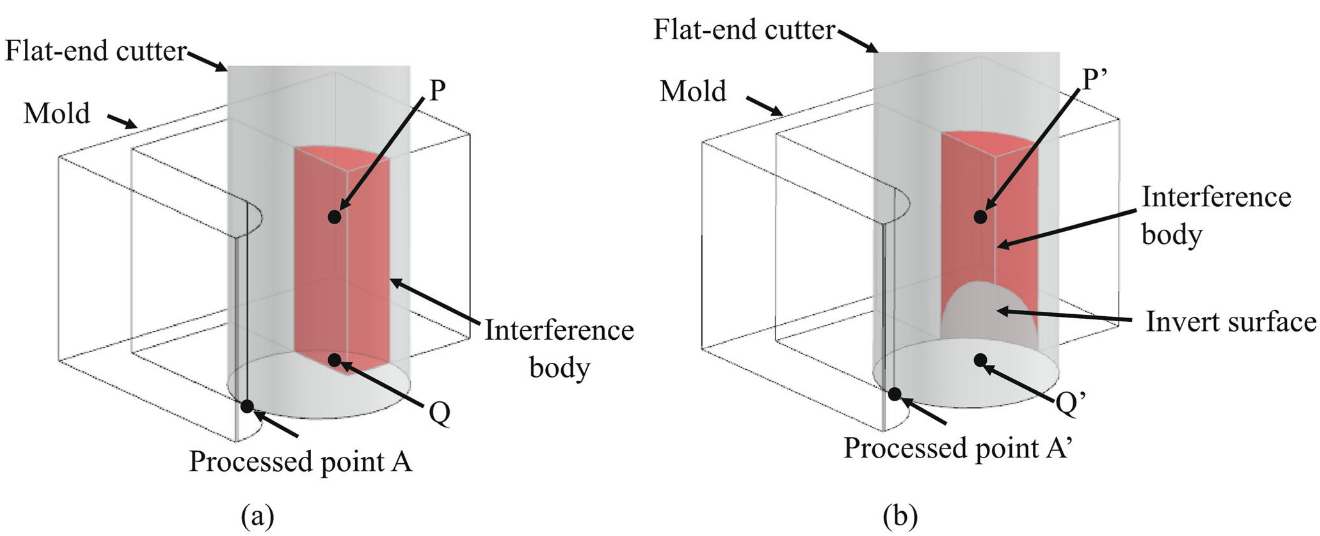


Fig. 6 **a** A mold without invert surface. **b** A mold with invert surface

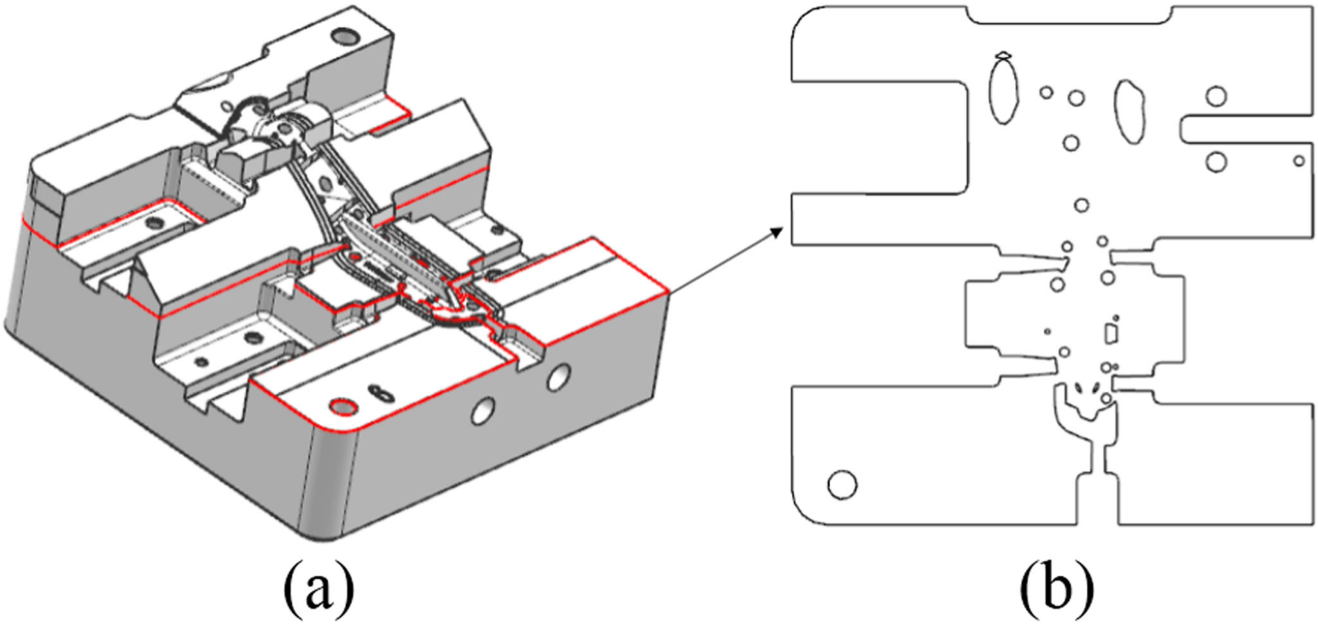


Fig. 7 **a** Mold to detect uncut regions. **b** One layer of section curves

shown in Fig. 6 b. If the bottom edge of the flat-end cutter cannot intersect with the mold as shown in Fig. 6 b, the point Q′ is outside of the mold. It means that the invert surface exists on the mold which obeys the precondition. As a consequence, the bottom edge of the flat-end cutter must intersect with the mold. The sufficiency can be proved.

If the bottom edge of a flat-end cutter intersects with the mold, thus, this flat-end cutter can intersect with the mold. That is, the uncut region exists. The necessity can be proved.

In the first step of the toolpath simulation method, the mold as shown in Fig. 7 a is sectioned with a set of paralleled planes, and one layer of section curves is illustrated in Fig. 7 b. According to Theorem 1, the bottom edge of a flat-end cutter can be employed to detect uncut segments in those section curves. Practice has proved that those uncut segments could be used to locate the uncut regions, as long as the gap distances between these section planes are smaller than the radius of that cutter. The detected uncut segments vary with the dimension of flat-end cutters. The smaller the radius of the cutter is, the smaller the uncut regions are. Therefore, only that flat-end cutter, whose radius is the smallest in the cutter library, is employed to detect the uncut segments.

When a point of the detected curve is processed by the bottom edge of the flat-end cutter, the interference between the bottom edge and the detected curve itself is called *self-interference* as shown in Fig. 8a; otherwise, the interference between the bottom edge and other section curves are called *other-interference* as shown in Fig. 8b. The uncut segments resulted from the self-interference should be detected first; otherwise, the uncut detection with other interference may result in bad consequence (as addressed in Sect. 4.1.2). The curve offset method is an effective way to detect self-interference, but it cannot handle the other-interference. In the other-interference, the detection is handled in two steps. All the

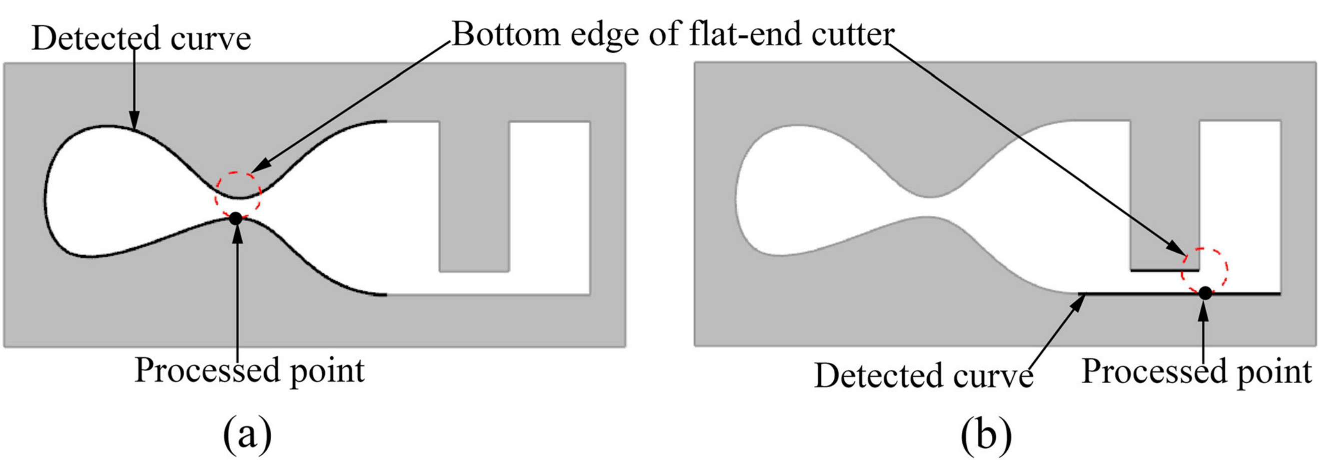
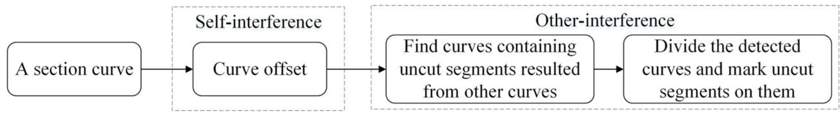


Fig. 8 **a** Detect uncut segments resulted from self-interference. **b** Detect uncut segments resulted from other-interference

Fig. 9 Procedure of uncut detection with the flat-end cutter



section curves containing uncut segments will be found in the first step. Then, how other curves can influence that machinability of the detected curve will be figured out, and the uncut segments on it are found. The procedure of uncut detection with the flat-end cutter is illustrated in Fig. 9.

4.1.1 Uncut detection in the self-interference

In the detecting process, the self-interference can be classified into two types: local and global interference. In 2D curves, the local interference occurs if and only if the maximum curvature value of the detected curve is bigger than that of the cutter [20, 21]. Otherwise, when the minimum distance from the offset curve to the detected curve is less than the radius of the cutter, the global interference occurs.

The curve offset method, an efficient way to detect either local or global interference, is applied into this method. The offset of a section curve, whose offset distance is equal to the radius of the flat-end cutter, is the trajectory of the cutter center. The self-intersection of an offset curve indicates that the uncut segment which resulted from self-interference exists. The segments of offset curves, whose minimum distances to the detected curve are less than the radius of the cutter, will be identified. Thus, the segments on the section curves, which are the correspondent segments of those identified segments, will be marked as uncut segments. When the local interference occurs, the uncut segments are as illustrated in Fig. 10 a, and Fig. 10 b shows the uncut segments when global interference occurs. In this way, the uncut segments caused by the self-interference can be detected. The method for fast calculation of the minimum distance between two 2D curves will be introduced in the next step.

4.1.2 Uncut detection in the other-interference

Finding all the curves containing uncut segments resulted from other-interference When a 2D curve is processed by a flat-end cutter, the trajectory of the bottom edge of the cutter is as shown in Fig. 11a. The outline of the trajectory is called the *P-region* of that curve (short for processing region). The *P*-region of a curve usually consists of four curves: the processed curve, its offset curve whose offset distance equals to the cutter diameter, and two semi-circles to make it enclosed.

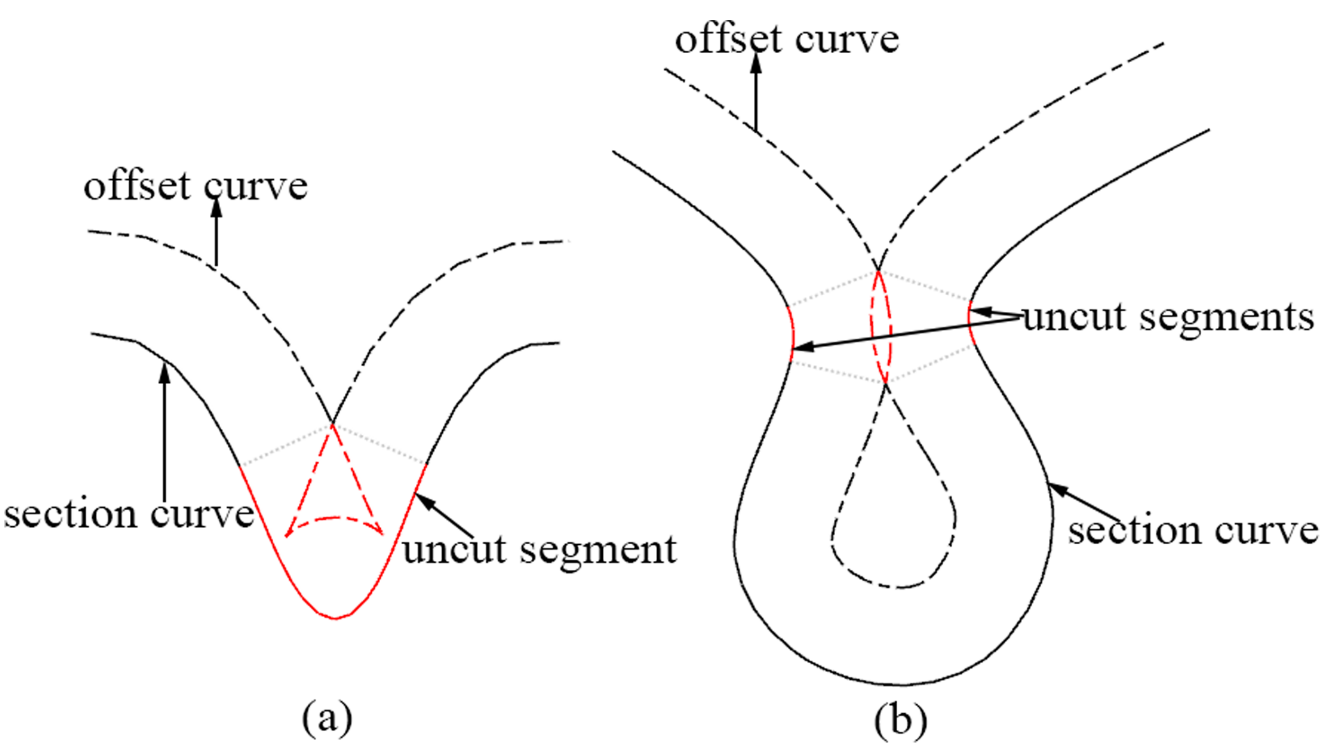


Fig. 10 **a** Local interference. **b** Global interference

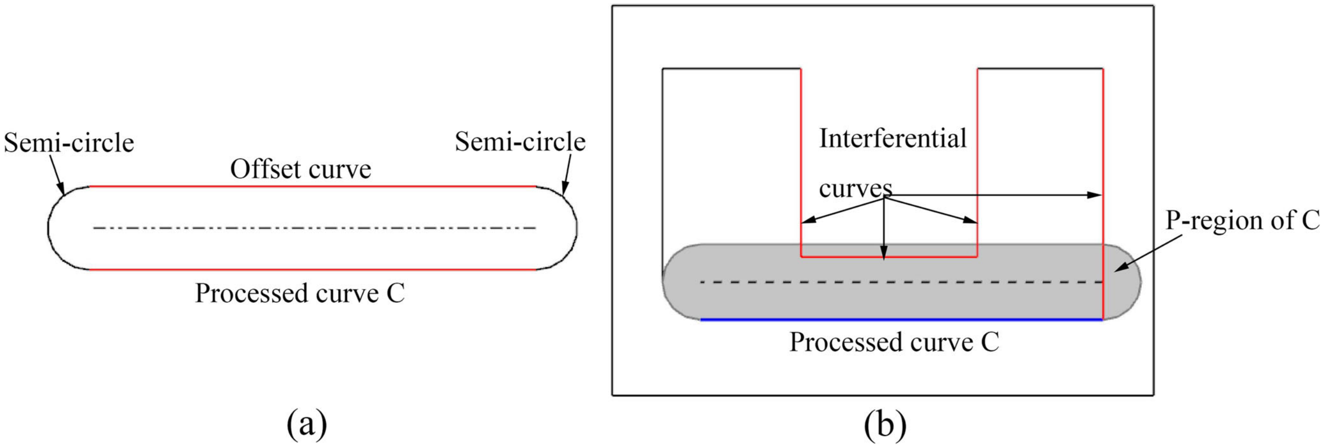


Fig. 11 **a** *P*-region of the processed curve. **b** Interferential curves of curve C

The *P*-region of a curve can help to distinguish whether this curve contains uncut segments resulted from other-interference. Obviously, if the *P*-region of a particular curve intersects with other section curves, that curve contains uncut segments that resulted from other-interference. Consider curve C in Fig. 11b as example. There are four interferential curves (red color) intersecting with the *P*-region of curve C; thus, these four curves can result in uncut segments on curve C. In this way, all the curves containing uncut segments can be detected. For one section curve, the interferential curves will also be marked. In the next step, the uncut segments resulted from each interferential curve will be marked.

In the toolpath simulation method, *P*-regions of all the section curves will not actually be created due to the enormous computation burden. The step of creating the *P*-region is replaced by an algorithm called *Minimum-distance* algorithm. This algorithm can judge whether the minimum distance between two curves is less than the radius of the cutter. Let curve C_0 and curve C_1 be the two curves in a layer of the section curves, and R_f be the radius of the flat-end cutter. The main motivation of the minimum-distance algorithm comes from that if C_0 can intersect with the *P*-region of C_1, then the

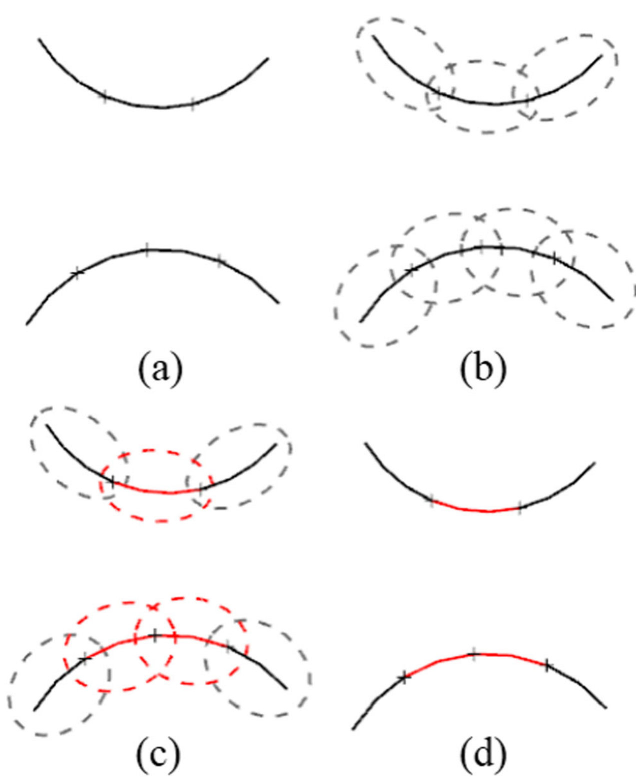


Fig. 12 Minimum-distance algorithm: **a** the division of both curves; **b** ellipse generated according to each segment; **c** quick location of the segments whose minimum distance may be less than *R*; **d** sampling segments to verify if their minimum distance is less than *R* or not

```
Algorithm 1: Judge if any interference occurs between two curves ▷ : comments
Input: C_A: Curve A;
       C_B: Curve B;
       R: the radius of cutter.
Output: I_f: If interference occurs
I_f ← false;
Let N_A = Count of segments of C_A;
Let N_B = Count of segments of C_B;
for (i = 0 → N_A − 1) do
    for (j = 0 → N_B − 1) do
        Let EAi = Ellipse generated with the segment i in C_A;
        Let EBj = Ellipse generated with the segment j in C_B;
        Let d_ij = distance between EAi and EBj;
        if (d_ij > R) then
            Continue;                                  ▷ There is no inference
        else
            Let P_i = the points sampled from segment i of C_A
            uniformly;
            Let P_j = the points sampled from segment j of C_B
            uniformly;
            Let dNew_ij = distance between P_i and P_j;
            if (dNew_ij > R) then
                Continue;                              ▷ There is no inference
            else
                I_f ← true;                            ▷ Inference occurs
                return I_f;
            end
        end
    end
end
return I_f;
```

Fig. 13 Minimum-distance algorithm

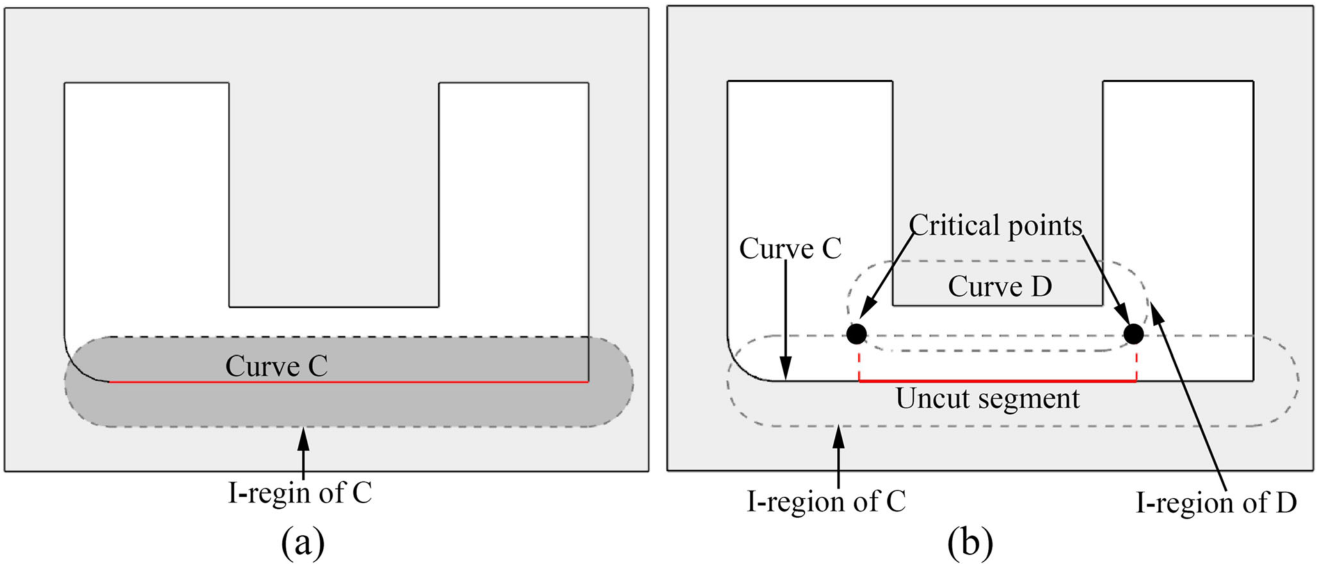


Fig. 14 **a** *I*-region of a section curve C. **b** Critical points of two *I*-regions

minimum distance between C_0 and the offset of C_1 is less than the radius of the flat-end cutter.

It is common sense to sample both curves into a large number of points, respectively, then the minimum distance between two groups of points can be used as a good approximate to the minimum distance between these two curves. It is computationally very expensive to calculate the minimum distances among the huge number of curves. The Minimum-distance algorithm divides both curves into several segments, and the minimum distance between any two segments on different curves can be calculated fast, thus improving the efficiency.

In the Minimum-distance algorithm, a curve with the length L is divided into n segments with the same length, and

$$n = \left\lceil \frac{L}{R_{\mathrm{f}}} \right\rceil \quad (1)$$

$\lceil\,\rceil$ denotes the ceiling operator. After division, the distance of two end points of each segment is less than R_{f} as shown in Fig. 12 a. Let the two endpoints of each segment be the two focal points of an ellipse, and R_{f} be the length of this ellipse principal axis as shown in Fig. 12 b. The segment must be within its corresponding ellipse. For any pair of segments on different curves, if the minimum distance of their corresponding ellipses is larger than R_{f}, the *P*-region of each segment will not intersect. Thus, only those pairs of segments as shown in Fig. 12c (red color), whose minimum distances of their corresponding ellipses are less than R_{f}, will be used to verify if their minimum distance is less than R_{f} or not. Those pairs of segments will be sampled uniformly into two sets of points, and the minimum distance between the two sets of points will be verified if their *P*-regions intersect, as shown in Fig. 12d (red color).

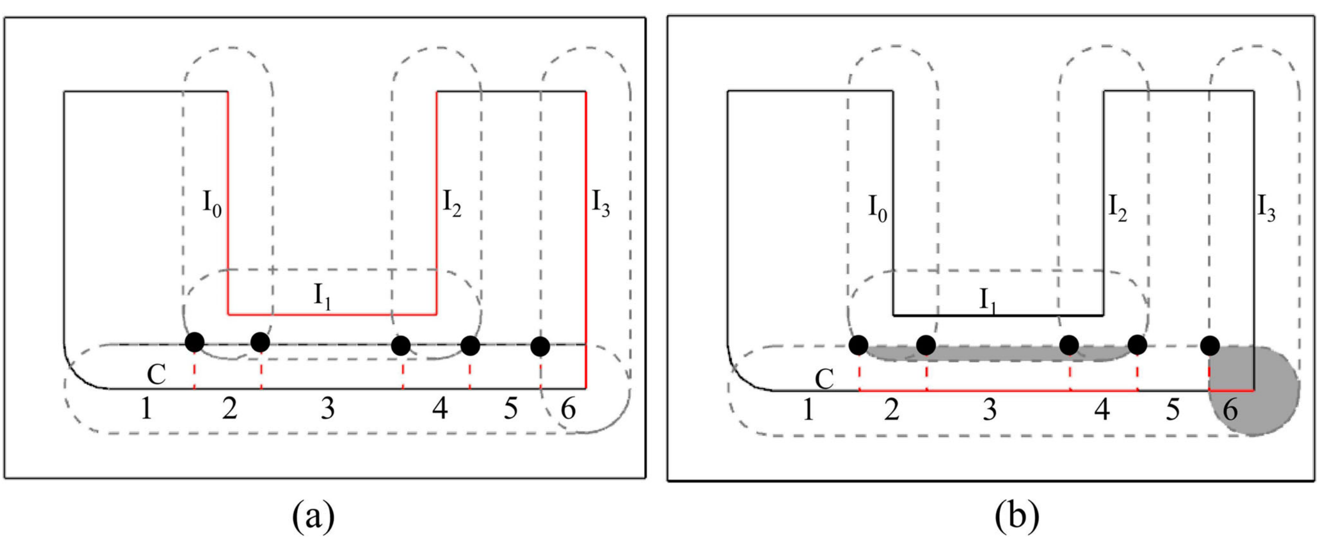


Fig. 15 **a** *I*-regions of interferential curves. **b** Uncut segments caused by curve C

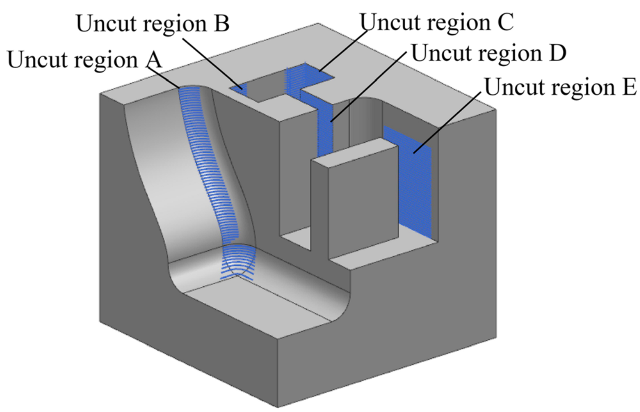


Fig. 16 Uncut regions detected with the flat-end cutter

Thus, the Minimum-distance algorithm can be used to effectively check if the minimum distance between two curves is less than R_f or not, on account that it is much easier to calculate the minimum distance of two ellipses. The Minimum-distance algorithm is fast and precisely. The procedure of the Minimum-distance algorithm is summarized in Fig. 13.

Divide the detected curves and mark uncut segments on them How other curves influence the machinability of the detected curve should be figured out. For any detected curve, there must exist a region. Once the cutter's bottom center is located in that region, the cutter will intersect with that detected curve. The region is called *I-region* (abbreviation for interferential region) of that curve, and the boundaries of I-region can be expressed as

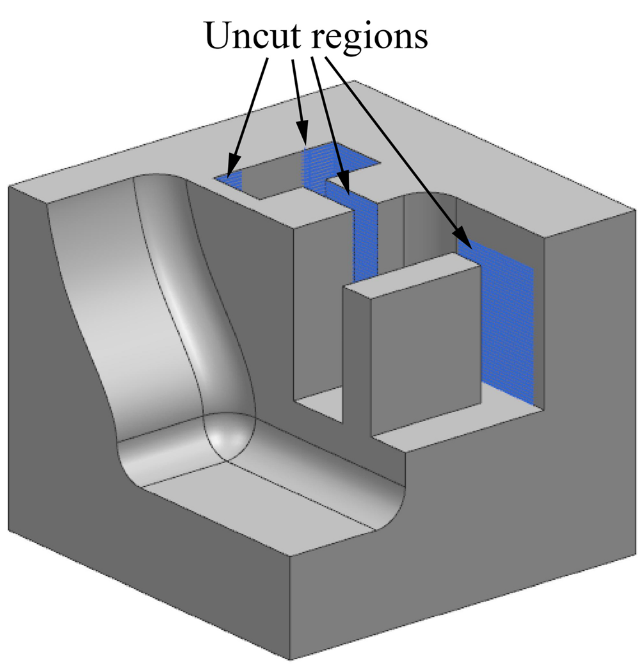


Fig. 17 Uncut regions detected by the flat-end cutter and the ball-end cutter

$$\begin{cases} d(t) = d_0(t) \pm N'(t) \bullet R \\ |\overrightarrow{P} - \overrightarrow{P}_0| = R \\ |\overrightarrow{P} - \overrightarrow{P}_1| = R \end{cases} \tag{2}$$

where $N'(t)$, P_0, and P_1 are the normal vector, start point, and end point of a section curve $d_0(t)$ respectively. Take curve C in Fig. 14 a as example. The *I*-region of curve C is different from the *P*-region of a section curve illustrated in Fig. 11 a. The self-intersection of an offset curve usually results in the failure of generation of *P*-region and *I*-region, and this is why self-interference should be detected first.

The crossing point of *I*-regions of two curves, called *critical point*, has the same distances to the two curves. Those critical points inside the mold are ignored of course. Consider the case in Fig. 14b; the *I*-regions of curve C and curve D have two critical points, and these two critical points can divide the curve C into three segments. Obviously, the middle segment (red color) is uncut. In this way, all the uncut segments can be detected.

Consider the curve C shown in Fig. 14 as instance. The section curve C has four interferential curves I_0, I_1, I_2, and I_3. Figure 15 a shows that the *I*-regions of curve I_0, I_1, I_2, and I_3 have five crossing points with the *I*-region of curve C. Therefore, curve C is divided into six segments. When the segments 2, 3, 4, and 6 are processed, the interferences will occur; therefore, the segments 2, 3, 4, and 6 are marked as uncut as shown in Fig. 15b.

In the same way, all the uncut segments can be detected. Consider the case in Fig. 16. There are five uncut regions on the mold. Obviously, the uncut region A can be machined by the ball-end cutter. This problem will be discussed in Sect. 4.2.

4.2 Uncut detection with the ball-end cutter

4.2.1 Theorems on uncut detection with the ball-end cutter

Theorem 2 *Surface* **r** *is free of local interference, and* $\mathbf{r_i}$ *represents all surfaces around* **r***;* **r** *and* $\mathbf{r_i}$ *are* C^0*, piecewise* C^2*. If all points on the boundaries of surface* **r** *do not interfere with surface* $\mathbf{r_i}$*, then none of the points on surface* **r** *interferes with surface* $\mathbf{r_i}$.

Proof This theorem has been proved. For a comprehensive review, the readers may refer to reference [21] for details.

Theorem 3 *For a mold without invert surfaces, the necessary and sufficient condition for existence of uncut area of ball-end cutter is that the spherical surface of the ball-end cutter intersects with the mold.*

Proof This theorem is similar to Theorem 1, and the readers may refer to Sect. 4.1.

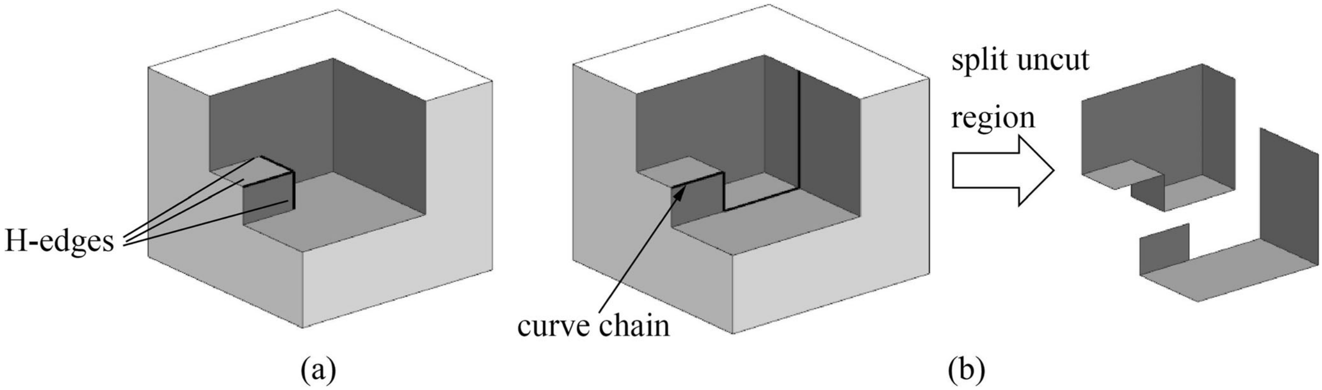


Fig. 18 **a** H-edges in the uncut regions. **b** The H-edges are used to form a curve chain to split invalid region into valid sub-regions

4.2.2 Process of uncut detection with the ball-end cutter

Those uncut regions detected with the flat-end cutter will be checked whether they can be machined by ball-end cutter. With Theorem 2, if all points on the boundary of surface $\mathbf{r}$ are machinable and do not interfere with surface $\mathbf{r_i}$, no point on surface $\mathbf{r}$ interferes with surface $\mathbf{r_i}$, regardless of whether there is a local interference region inside surface $\mathbf{r_i}$. Therefore, the interference between the cutter and mold can be detected with the surface boundaries [21].

There is no need to detect every face in the uncut region. The surfaces, whose maximum curvature is bigger than that of the ball-end cutter, are called *curved faces*. Obviously, only the curved faces in the uncut regions need to be detected.

In uncut detection with the ball-end cutter, the boundaries of the curved face are sampled first. According to Theorem 3, when a point is processed by a ball-end cutter only if the spherical face of the ball-end cutter does not interfere with the mold, the point is machinable, otherwise unmachinable. Therefore, the cutter center point P is calculated at each sampled point, and the minimum distance between point P and all the other surfaces of mold is calculated. If all the sampled points are machinable, the curved surface is machinable; therefore, it is removed from the uncut regions. If not, the curved surface is still uncut. Consider the mold in Fig. 16. After detection with the ball-end cutter, the final uncut regions are shown in Fig. 17.

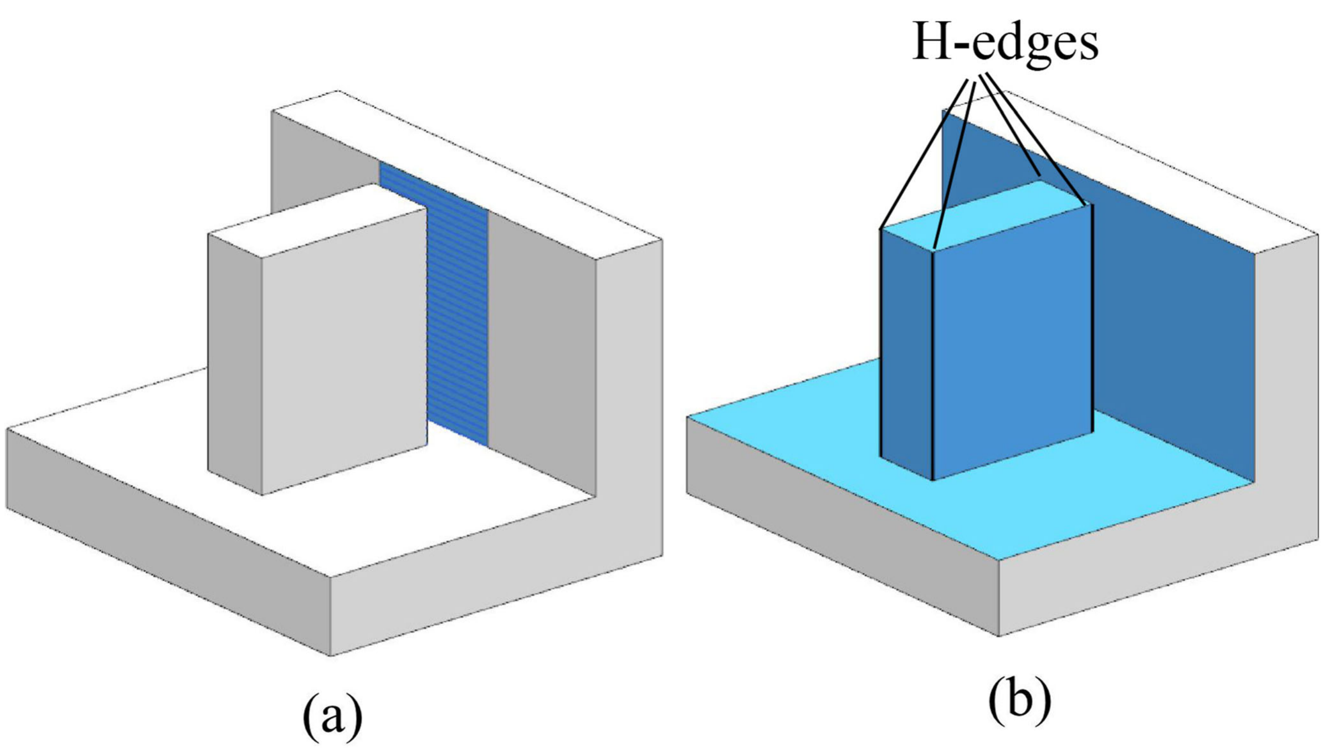


Fig. 19 **a** Uncut region detected in the toolpath simulation method. **b** Uncut region detected by the H-edge-based algorithm

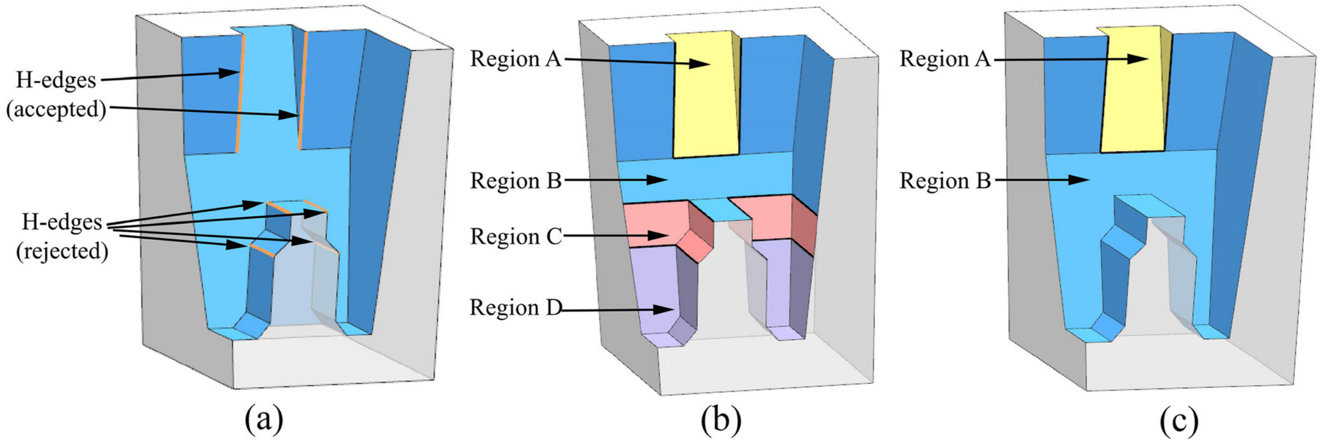


Fig. 20 The reduction of H-edges

4.3 Automatic division of uncut regions

After uncut detection with the flat-end cutter and ball-end cutter, the uncut regions of a mold are presented as many scattered pieces. Theoretically, each region should design an electrode to process this uncut region. In most cases, the electrode itself is not machinable for the existence of sharp corners or concave edges. It is common practice to split the uncut region into sub-regions until all the electrodes designed for the sub-regions are machinable.

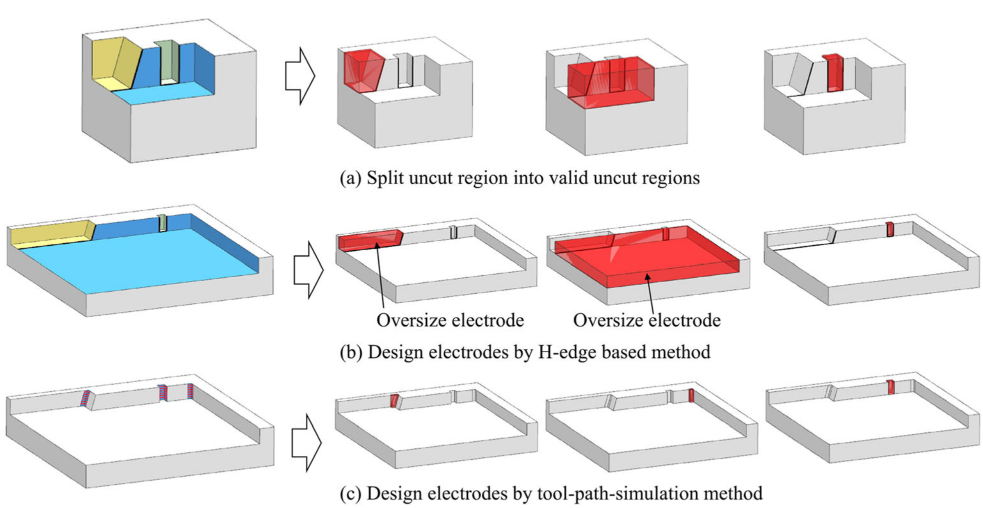
Two methods are usually employed to split uncut regions. Zhou et al. [21] split the uncut regions based on volume decomposition. The volume decomposed-based algorithm employs the planes to divide the uncut regions; therefore, it is not good at handling with those uncut regions containing curved surfaces. Another method, the H-edge-based splitting algorithm proposed by Lee et al. [8], is more flexible in the splitting process.

The uncut region in Lee's opinion is a set of faces such that (i) at least one face f_e in R requires EDM and (ii) for any faces f_i in R, there exists at least one other face f_j in R that shares a concave edge with f_i. The convex edges causing machining problems in the uncut region are called H-edges as shown in Fig. 18 a. The H-edges are used to form a curve chain to divide the invalid region into valid sub-regions as shown in Fig. 18 b, and thus to design electrode tips.

The H-edge-based algorithm is powerful to split the uncut regions precisely. This paper also employs the algorithm, but improves it in several aspects.

(1) In the H-edge-based algorithm, the faces in the uncut regions can only be classified into two types: the whole face can be processed and the whole face cannot be processed. Those faces, part of which can be processed, are recognized that the whole face can be processed, thus increasing the uncut area and count of H-edges. A huge number of H-edges makes it complicated to form the s-chain so that the algorithm is not efficient in some cases. In the toolpath simulation method, the uncut regions are more accurate and smaller; thus, the count of H-edges is reduced and the splitting process is also much easier. Consider the example shown in Fig. 19. The uncut area

Fig. 21 **a** Normal size uncut region and electrodes. **b** Oversize electrodes designed by the H-edge-based algorithm. **c** Electrodes designed by the toolpath simulation method



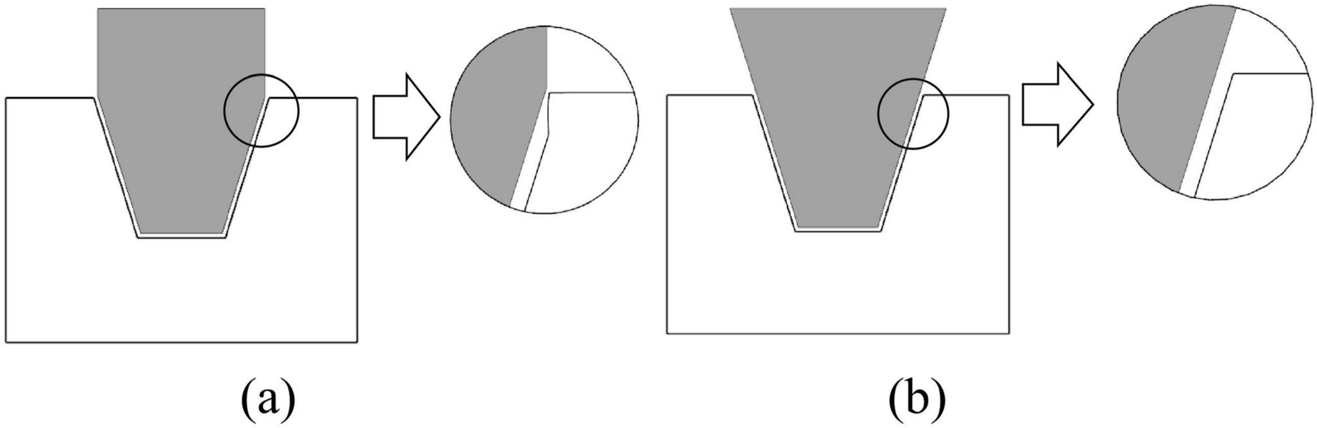


Fig. 22 a Problem of leftovers. b Electrode refinement

(blue area) in Fig. 19a, detected by the flat-end cutter with the toolpath simulation method, is smaller than the uncut area (blue area) in Fig. 19b, which is detected by the H-edge-based algorithm. There are no H-edges in Fig. 19a, but four H-edges in Fig. 19b. The decrease in H-edges shows the high efficiency of the toolpath simulation method.

(2) Actually, a set of concave edges on the electrodes could be processed by wire EDM, such that (i) the profile of the edge is a horizontal line and (ii) the adjacent faces of the edge both are planes. The corresponding convex edges in the uncut regions are no longer H-edges in the toolpath simulation method, thus reducing the count of H-edges. Consider the case in Fig. 20a. There are six H-edges in the uncut region, but four H-edges meet the conditions above and they need to be rejected. Figure 20b shows the splitting process of six H-edges, and Fig. 20c shows the splitting process of 2 H-edges. The splitting process of 6 H-edges is much more complex than the splitting process of two H-edges.

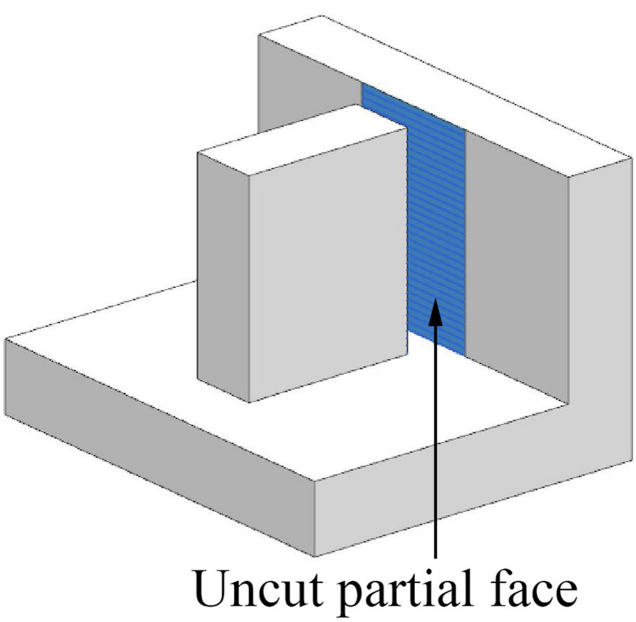


Fig. 23 Partial face is uncut

(3) The common problem in Lee and Zhou's method is that the size of electrodes usually is too bigger to be used. In their opinions, no matter how big the size of an uncut region is, if only the shape of the region is kept, the division of that region is in the same way. This is because the splitting process proposed by them did not take the size of the uncut region into consideration. Consider the uncut region in Fig. 21 a, the size of the uncut region and electrodes is normal. But if the uncut region is enlarged in Fig. 21 b, the electrodes designed by the H-edge-based algorithm are too large to be machined. On the contrary, the uncut regions detected with the toolpath simulation method are rather reasonable as shown in Fig. 21 c. This is because the uncut regions are detected more precisely.

5 Design electrodes for uncut sub-regions

A solid model of the electrode for each valid sub-region is constructed by a set of CAD modeling operations. There are two problems needed to be solved in order to generate a qualified electrode. Firstly, if the discharging faces of the electrode are exactly the same as the sub-region, the electrode is unqualified for the leftovers as shown in Fig. 22 a. In order to avoid the problem, the marginal discharging faces on the electrode must be bigger than the corresponding faces on the mold as shown in Fig. 22b. Secondly, the existence of partial faces in sub-regions

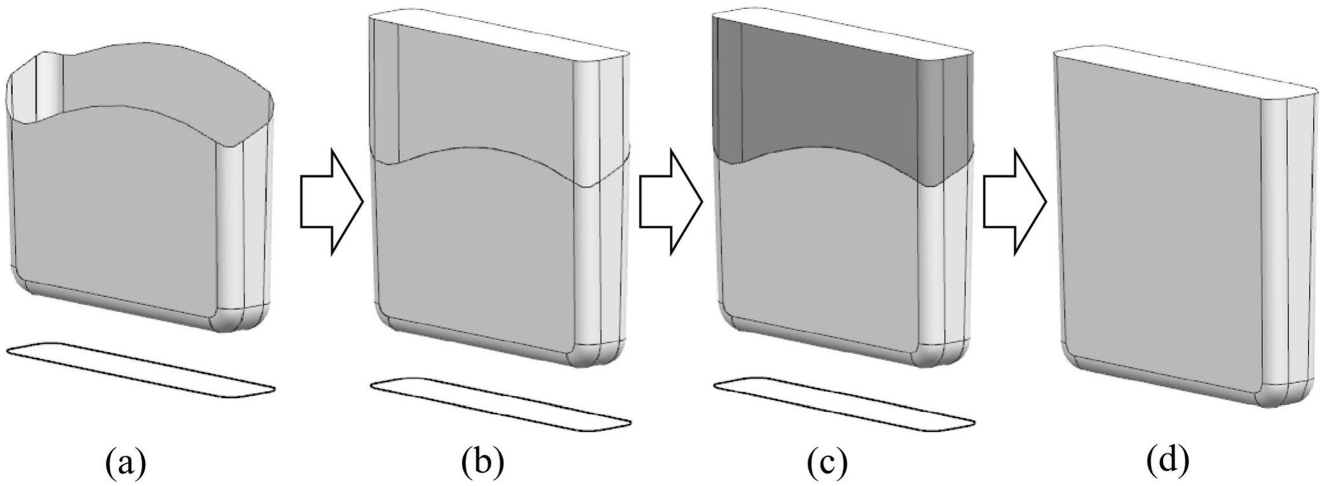


Fig. 24 Process of extruding method: **a** project boundaries, **b** trim-extruding feature, **c** find face to extend, **d** extend faces to the top of the electrode tip

will make it hard to generate an appropriate electrode as shown in Fig. 23. In order to solve the two problems, two different ways to generate electrodes are specially introduced for certain instances, called as extruding method and cutting method.

5.1 Extruding method for generating electrode tips

For the sub-region excluding partial faces of the mold, the extruding method is appropriate to generate the electrode. In the extruding method, the region boundary of an uncut region is first projected onto a plane above the region along the z direction to form the projected boundary as shown in Fig. 24a. The projected boundary is extruded along -z to form a solid body so that all of the faces in the region are included in the solid body. The solid body is then trimmed by the region to form the electrode illustrated in Fig. 24b. For overcoming the problem of leftovers, the adjacent faces of the top face need to be extended to the top of the electrode illustrated in Fig. 24 c

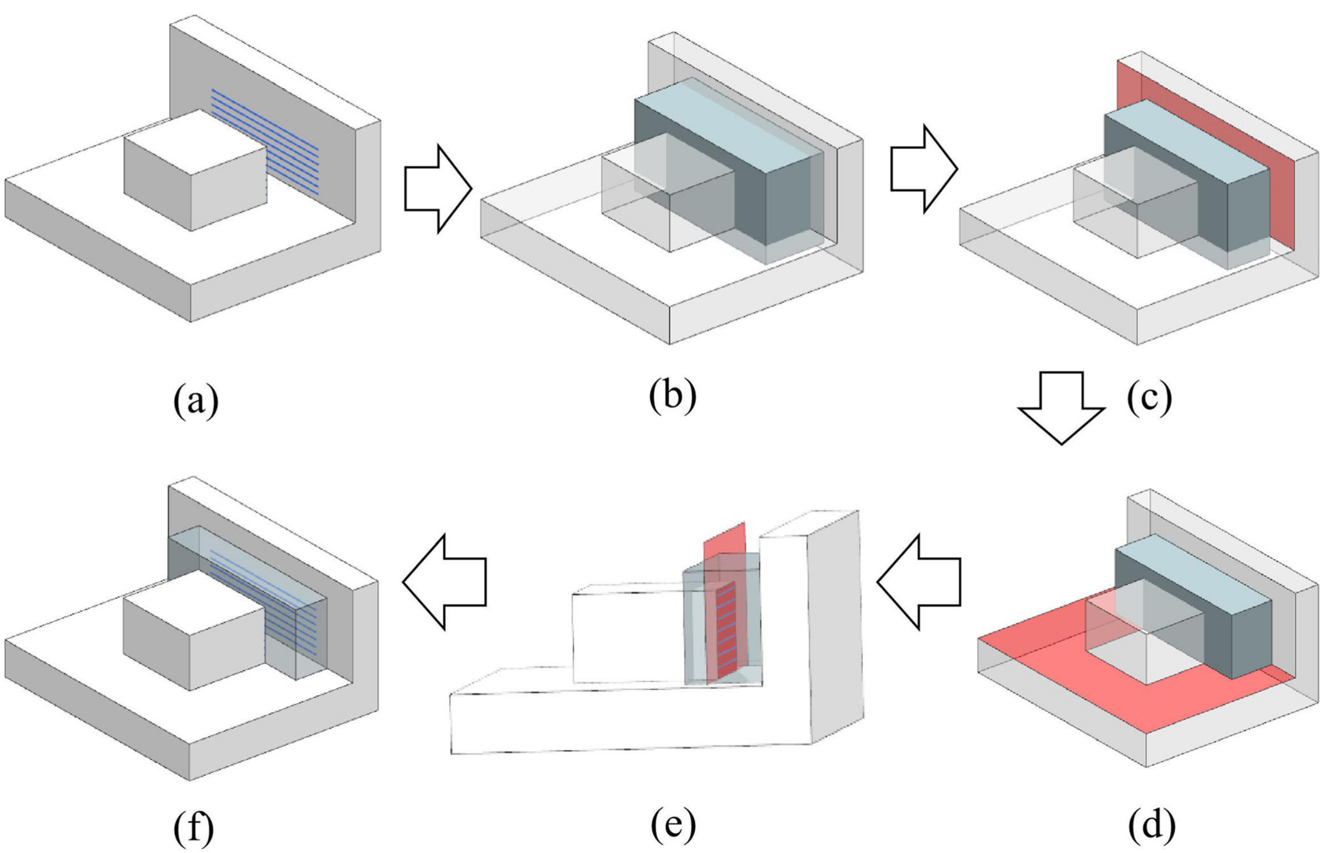


Fig. 25 Process of the cutting method

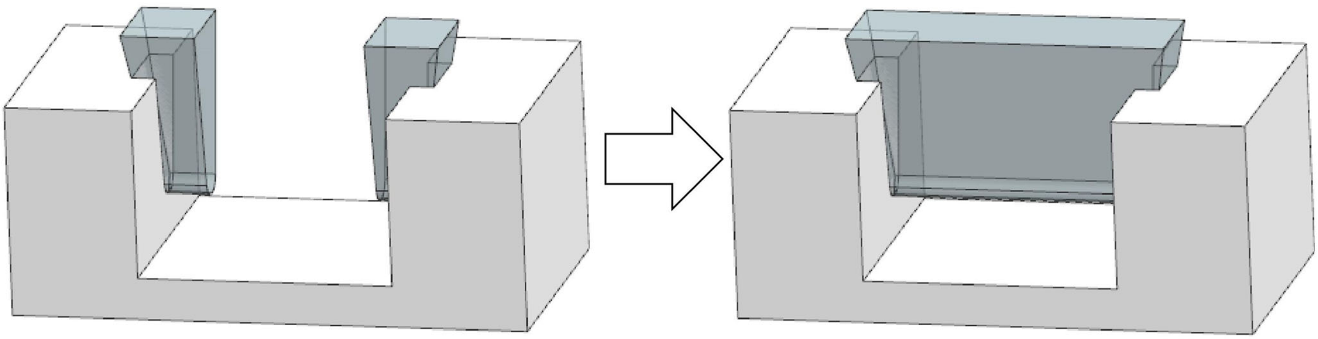


Fig. 26 The combination of two electrodes

and d. This method can solve the problem of leftovers, and the physical electrodes generated for the EDM process are qualified.

5.2 Cutting method for generating electrode tips

In most cases, the uncut sub-region contains partial faces. Therefore, the exact boundary of the uncut sub-region is hard to be generated and the extruding method cannot work well. Thus, the cutting method for the design electrode tip is developed to avoid the generation of project curves.

In this approach, the minimum bounding box of the uncut region is created; what is more, each face of the bounding box will be offset out a certain value to avoid the problem of leftovers. The bounding box will be trimmed by all the enlarged faces in the uncut region. The process is as shown in Fig. 25.

5.3 Electrode refinement and holder generation

In addition to the toolpath simulation algorithm for uncut detection and the electrode tip designing algorithm, an electrode refinement method is also implemented to guarantee the promotion of the quality of physical electrodes. The refinement method of combining two electrodes is introduced. As known, the clamping of electrodes will lead to location errors. This electrode refinement method combines two or more electrode tips together, and those electrode tips can process the mold for just one time, thus eliminating the location errors and improving the machining precision. But only those electrodes, whose shapes and sizes are totally the same, can be combined. The example in Fig. 26 can illustrate the process of this refinement.

The electrode holder is used to hold the electrode tip during CNC and EDM machining. Standard-sized electrode holders should be used whenever possible; thus, standard fixtures can be used, which will greatly reduce the setting time in CNC and EDM. A library containing standard-sized electrode holders is constructed in this module. The electrode holder is positioned at the top of the electrode tip, while the center coincides with each other in the X and Y directions. Two or more electrode tips should share the common holder on condition that the minimum distance is within certain limits as shown in Fig. 27. The certain limits are usually defined by the industries.

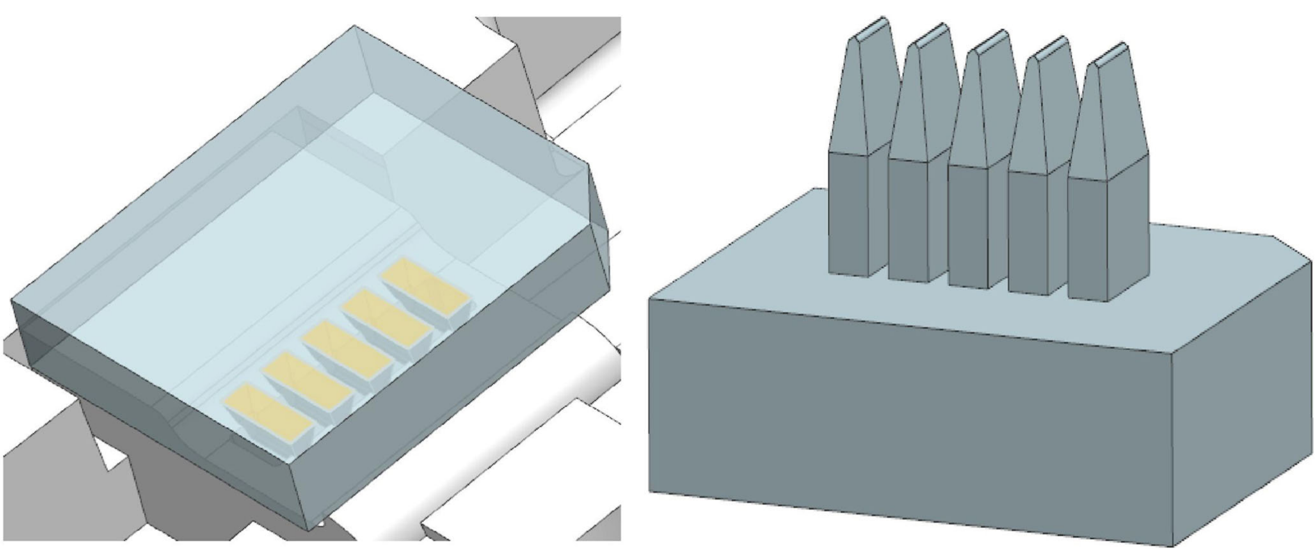


Fig. 27 Five electrode tips share the same holder

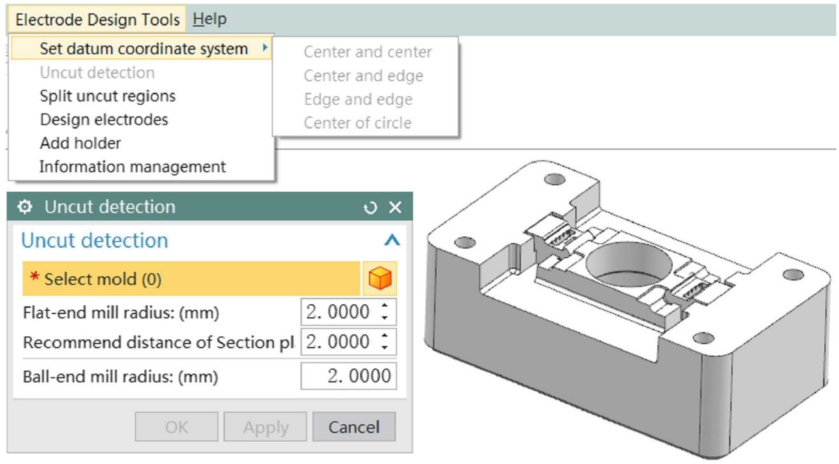


Fig. 28 Menu and interface of the intelligent CAD system

6 System implementation and case study

The proposed techniques for the uncut detection and automation of the electrode design are implemented as an intelligent CAD system using the C ++ programming language and are built as dll files. The intelligent system is integrated with the Unigraphics NX 10.0 system through its NX OPEN C++ Interface. The input to the intelligent system includes a geometric model of a mold, the library of cutters, and the directions for EDM operation. The outputs of the intelligent system are the electrodes and uncut segments in the Unigraphics NX 10.0 system, and thus, the electrodes can be used in the manufacture of electrodes in the following steps.

The intelligent CAD system consists of the 6 tools, and the menu of the system is shown in Fig. 28: (1) Set datum coordinate system for mold (work piece). (2) Uncut detection. After the types and radii of cutters are specified, the toolpath simulation algorithm is then invoked to section the mold and searches for the uncut regions. All the uncut segments marked in red color are illustrated as shown in Fig. 29 a. (3) Split uncut regions. In all the scattered uncut regions, the H-edges in the scattered uncut regions are detected and the curve chains are formed to divide those uncut regions. (4) Design electrode tips. Two methods are employed to design electrode tips automatically as shown in Fig. 29 b. (5) Add holders. The electrode tips, which match the conditions about distance and shape, will share the same holder. The typical electrodes are

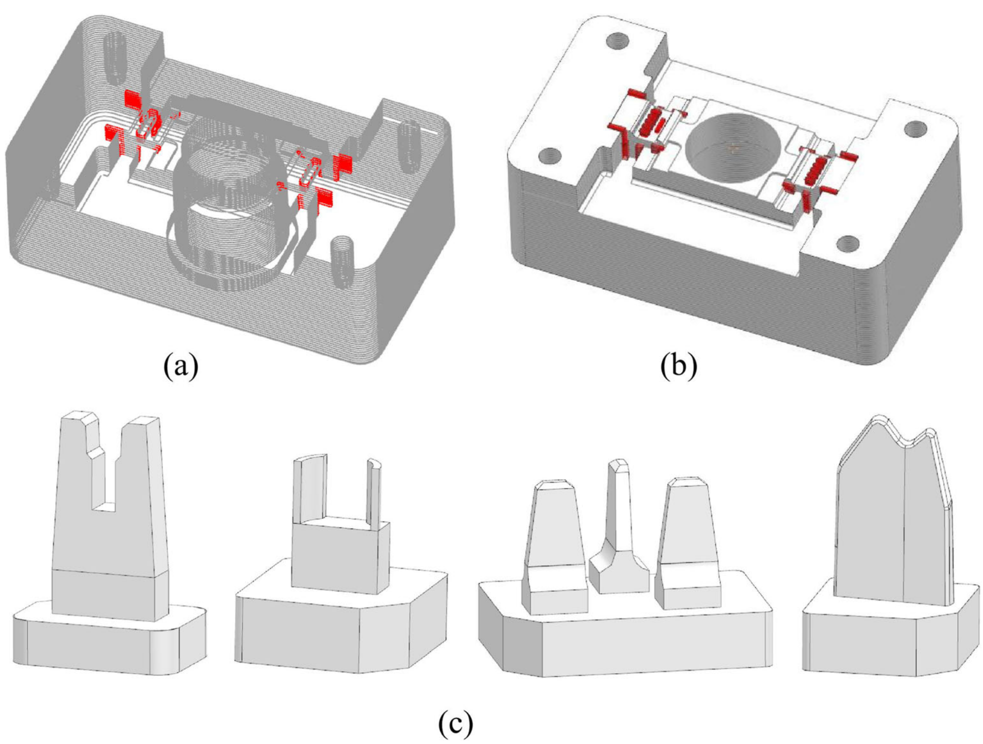


Fig. 29 **a** Mold to detect uncut regions. **b** All the section curves with uncut segments. **c** The electrode tools generated from the uncut sub-regions. **d** All the electrodes with a holder

Table 1 Performance evaluation

Performance data (min)	Uncut detection	Design electrodes	Uncut area check	Refine electrodes	Total	Total manual
Without intelligent CAD tool	5 (manual)	35 (manual)	3 (manual)	6 (manual)	49	49
With intelligent CAD tool	5	4	0	4 (manual)	13	4
Savings (%)	0.0	88.6	100	33.3	73.5	91.8

shown in Fig. 29 c. (6) Information management. After all, the information will be added into each electrode in forms of attributes for the following step of machining of electrodes.

The intelligent CAD system has been tested with 200 real design cases to verify its ability to address common geometries in the electrode design. A case study is presented to illustrate the major steps of the toolpath simulation method and evaluate the performance of the intelligent CAD system.

The performance of the design process is evaluated in terms of the time spent on the entire design process. An average engineer is asked to design electrodes with NX 10.0; in the meantime, another average engineer employs the intelligent CAD system to design the same electrodes. The performance data of these two engineers are shown in Table 1. The time spent can be divided into four parts: time to detect uncut regions, time to design electrodes, time to check the uncut regions where design electrodes are forgotten, and time to refine electrodes. Although most electrodes are implemented in the intelligent tool, the engineers may want to combine the electrodes or improve the quality of electrodes.

As shown in the last column of Table 1, the savings obtained by the intelligent CAD system can reach up to 73.5%, even to 91.8% when only the manual time is taken into consideration. The savings in design time are of great help to raise productivity. The savings in time indicate that the efficiency of the electrode design process can be improved by at least 70%, thus saving a large amount of time. The reduction of time can lower the price of consumers and make them more competitive in the market.

7 Conclusion

The process of uncut detection and electrode design takes up a substantial amount of time in the design and manufacture process of a plastic injection mold. In this research, an intelligent CAD system has been developed to automate the process. An innovative technique for detection of an uncut region and generation of electrodes overcomes the major limitations of existing methods. The innovation in our work is concluded as follows.

A toolpath simulation method to detect uncut regions on injection mold is proposed. This method can simulate the toolpath and detect the uncut regions precisely without generating the toolpath actually. This toolpath simulation method can also accurately detect and divide a single face into the uncut region and machinable region and show all the uncut segments to avoid missing any electrode design, instead of identifying the whole face as either machinable face or uncut face as in other methods, which is of great importance to reduce reworking and is very welcome among the electrode engineers. In this sense, the uncut region recognition is the most precise in all the methods.

The process of uncut detection and electrode design is free of human–computer interaction, thus saving a lot of time, lowering production cost, and improving the quality of electrode design. The saving in time costs obtained by the intelligent CAD system can reach up to 73.5%, even to 91.8% when only the manual time is taken into consideration.

However, there are still improvements to be achieved. The future work can focus on the integration of CAD/CAM/CAPP/CAI system of electrode design and manufacture.

References

1. Ho K, Newman S (2003) State of the art electrical discharge machining (EDM). Int J Mach Tools Manuf 43(13):1287–1300
2. Heo EY, Kim DW, Lee JY, Lee CS, Chen FF (2011) High speed pocket milling planning by feature-based machining area partitioning. Robot Comput Integr Manuf 27(4):706–713
3. Tapie L, Mawussi B, Bernard A (2012) Topological model for machining of parts with complex shapes. Comput Ind 63(5):528–541
4. Jha B, Ram K, Rao M (2011) An overview of technology and research in electrode design and manufacturing in sinking electrical discharge machining. J Eng Sci Technol Rev 4(2):118–130
5. Mognol P, Rivette M, Jegou L, Lesprier T (2007) A first approach to choose between HSM, EDM and DMLS processes in hybrid rapid tooling. Rapid Prototyp J 13(1):7–16
6. Ding X, Fuh J, Lee K, Zhang Y, Nee A (2000) A computer-aided EDM electrode design system for mold manufacturing. Int J Prod Res 38(13):3079–3092
7. Kuzman K, Nardin B (2004) Determination of manufacturing technologies in mould manufacturing. J Mater Process Technol 157:573–577
8. Lee Y, Li C (2009) Automation in the design of EDM electrodes. Comput Aided Des 41(9):600–613
9. Ma H, Zhou X, Wei L, Li J, Qiang N, Kong C (2018) A feature-based approach towards integration and automation of CAD/CAPP/CAM for EDM electrodes. Int J Adv Manuf Technol 98(9–12):2943–2965
10. Quang NH, Lin AC (2013) Determination of parting lines and multiple-undercut boundaries for complex CAD parts with application to multi-piece molds design. Appl Mech Mater 263-266:1633–1637

11. Wang G, Shan Y, Rongren WU (2006) Compensation of electrode orbiting in electrical discharge machining based on non-uniform offsetting. J Zhejiang Univ 45(14):1628–1634
12. Sriani T, Prihandana GS, Aoyama H (2015) Electrode design for orbiting EDM: an experimental approach. Appl Mech Mater 758: 159–163. https://doi.org/10.4028/www.scientific.net/AMM.758.159
13. Lo JS, Jiang CT (2019) Compensation method for profile deviations caused by the complex shape of electrodes in orbital electrical discharge machining. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-019-03601-9
14. Ding X, Fuh J, Lee K (2002) Computer aided EDM electrode design. Comput Ind Eng 42(2):259–269
15. Ding X, Lu Y, Fuh J, Lee K (2004) Optimal cutter selection for complex three-axis NC mould machining. Int J Prod Res 42(22): 4785–4801
16. Mahajan K, Knoppers G, Oosterling J, Van Luttervelt C (2004) Knowledge based design of EDM electrodes for mould cavities pre-machined by high-speed milling. J Mater Process Technol 149(1):71–76
17. Zhou C, Zhang S (2010) Automatic EDM electrode splitting based on volume decomposition. 2010 Second International Conference on Computational Intelligence and Natural Computing. https://doi.org/10.1109/CINC.2010.5643900
18. Geng W, Liu H (2018) A novel approach of automatically designing EDM electrodes for machining uncut regions. Comput-Aided Des Applic 15(4):1–11
19. Lee K, Kim TJ, Hong SE (1994) Generation of toolpath with selection of proper tools for rough cutting process. Comput Aided Des 26(11):822–831
20. Ding X, Fuh JY, Lee KS (2001) Interference detection for 3-axis mold machining. Comput Aided Des 33(8):561–569
21. Zhou C, Liang J, Zhang S (2009) A fuzzy approach to determine processing method in mould manufacturing. 2009 Second International Conference on Intelligent Networks and Intelligent Systems. https://doi.org/10.1109/ICINIS.2009.119