Recognition and Simplification of Holes in CAD Models of an Injection Mold for Mold Flow Analysis

Jiing-Yih Lai[1], Pei-Pu Song[2], Chia-Hsiang Hsu[3] and Yao-Chen Tsai[4]

[1]National Central University, Taoyuan, Taiwan, jylai@ncu.edu.tw
[2]National Central University, Taoyuan, Taiwan, scarlet811025@gmail.com
[3]CoreTech System (Moldex3D) Co., Ltd., Hsinchu, Taiwan, davidhsu@moldex3d.com
[4]CoreTech System (Moldex3D) Co., Ltd., Hsinchu, Taiwan, cloudtsai@moldex3d.com

Corresponding author: Jiing-Yih Lai, jylai@ncu.edu.tw

Abstract. In mold flow analysis, a mold base is usually modeled as a rectangular box to simplify the mold structure. However, in order to improve the accuracy of the analysis, the real, non-simplified model base must be considered in some applications. There are usually a large number of holes in an injection mold, and most of them pass through multiple plates. To convert the CAD models of an injection mold into solid meshes, some of the holes in the mold set need to be simplified. The purpose of this study is to recognize and simplify unnecessary holes in an injection mold for mold flow analyses. It is primarily divided into four steps: (1) hole recognition on each CAD model, including single and ladder holes, (2) recognition of related holes passing across different CAD models, (3) detection of holes that should be preserved, and (4) simplification of unnecessary holes. Three injection molds were used to demonstrate the feasibility of the proposed method.

Keywords: Hole Recognition, Hole Simplification, Feature Recognition, Feature Suppression, B-rep Model
DOI: https://doi.org/10.14733/cadaps.2020.88-107

1 INTRODUCTION

In mold flow analysis, it is necessary to convert the computer-aided design (CAD) models of an injection mold into solid meshes during the analysis. The main components of an injection mold that need to be modeled are the core, cavity, runner system, cooling channels, and the mold base. Traditionally, a mold base is modeled as a rectangular box to simplify the mold structure. However, in some applications, it is necessary to consider the non-simplified, original plates of the mold base in order to increase the accuracy of the mold flow analysis. As more components of the mold are involved and meshed, the total number of mesh nodes increases. Therefore, mesh reduction becomes an important issue to study.

Typical mold plates on a mold base are clamping plates, cavity plates, stripper plates, core plates, support plates, spacer blocks, locating plates, and ejector clamping plates. Holes are a common feature in these components. If all holes are preserved and meshed, then, because the mesh size near holes should be smaller, a tremendous amount of nodes would be generated. On the contrary, if most holes on mold plates are simplified, then the total amount of nodes can be substantially reduced. It can be noted that the simplification of holes on most mold plates would not affect the results of the analysis because they are assembled with counterparts during the injection process.

On the other hand, holes that need to be preserved in mold flow analyses sometimes go across different CAD models. For example, a cooling channel is usually composed of a series of holes passing through different mold plates, including both the core and cavity plates. Similarly, a runner system usually starts from the top plate, passes through a stripper or support plate, and finally reaches the core and cavity plates. Although hole recognition modules are available in some CAD systems, most of them focus on individual CAD models only; when a set of CAD models representing an injection mold are considered, each of them must be processed, one by one. To facilitate the automation of the CAD pre-processing process in mold flow analysis, it is necessary to detect and record all related holes across different CAD models, rather than within each CAD model only.

In feature recognition, most investigations employ the topological relationship of adjacent entities for the recognition of features. A method has been proposed, in accordance with the topological relationship of the boundary representation (B-rep) model, to solve the problem that the boundary of the holes is not filleted when using the attributed adjacency graph (AAG) method [8]. This proposed method utilizes the property that a hole, in the B-rep model for hole recognition, is always accompanied with an inner loop. However, the types of holes that can be recognized are limited. Song et al. [14] presented a method for the recognition and suppression of different kinds of holes on a mold base. An approach based on the loop data of the B-rep model was developed to search for holes on each CAD model. Holes that are connected in series on the same CAD model were also detected and recorded. Finally, a hole suppression algorithm was proposed to eliminate each set of connected holes. When a hole is filleted on its boundary, it may be necessary to recognize the fillet first, and then perform the recognition of the hole in terms of the fillet information. Several fillet recognition algorithms are available in literature [4], [7], [9], [17].

When a feature is located on multiple planes or surfaces, the feature recognition flowchart is more complex, as more topological relationships and information must be considered and evaluated. An algorithm that projects surfaces onto the projected plane, and then combines the volume decomposition method for the feature recognition, has been proposed [15]. This method was primarily used for the efficient planning of NC tool cutting paths. Surface features were recognized on an STL model by an alternative algorithm, which evaluates the curvature of triangular meshes [16]. To recognize and classify convex features in terms of the parameters of B-spline surfaces a different algorithm has been proposed and applied for sheet metal parts [19]. You et al. [18] isolated the surface features by eliminating all filleted features from the CAD model, and then established an algorithm in terms of the AAG graph for the recognition of the remaining surfaces. A method for the recognition of aircraft structural parts was developed in terms of the holistic AAG algorithm [10]. Several algorithms were presented to modify the operating parameters and to slice and filter the surfaces.

The feature data extracted from various feature recognition methods are generally not sufficient for solid mesh generation because they are primarily used to describe the feature shape. For the feature data to be useful for mesh generation, they must be designed and extracted in accordance with the needs of different meshing methods. Several approaches are available in the literature to decompose, extract, or simplify solid models to obtain data for automatic mesh generation. Chong et al. [3] focused on idealizing finite element models and proposed some operations to allow the user to reduce the dimensionality of geometric models automatically by using a decomposition and reduction method. Boussuge et al. [1] proposed an idealization approach based on generative shape processes to decompose solids. Instead of using an existing construction tree in CAD software, they provided a graph containing all non-trivial construction trees using generative processes, which is more useful to evaluate variations of idealization. Makem et al. [11] employed shape metrics generated using local sizing measures to identify long, slender regions within a thick body, and then proposed a procedure to partition the thick region into a non-manifold assembly of long, slender, and complex sub-regions. Both structured anisotropic and unstructured meshes can then be employed in different regions to obtain an efficient distribution of meshes. Juttler et al. [5] presented a technique for segmenting a solid model with an edge graph of only convex edges into a collection of topological hexahedra. An edge graph, defined by the sharp edges between the boundary surfaces of the solid, was employed to repeatedly decompose the solid into smaller solids until all of them belong to a certain class of predefined base solids. Each of the solids can then be meshed by hexahedral meshes.

The purpose of this study is to recognize all holes in an injection mold composed of a set of CAD models, and arrange the related holes that pass between different CAD models. A hole recognition algorithm has already been presented [14], which identifies single and ladder holes in a CAD model, where a ladder hole represents a set of holes that connect to each other in series. In this study, we focus on the recognition of a series of holes passing across different CAD models and arrange them in sequence. Holes that are used in mold flow analyses, such as core, cavity, runner system, and cooling channels, are identified and preserved, whereas the others in the mold base are automatically simplified.

2 OVERVIEW OF PROPOSED HOLE RECOGNITION AND SIMPLIFICATION METHOD

To aid the analysis, holes can be classified into several types. A single hole exists alone, such as the cases in Figures 1(a) and (b). Several holes that connect to each other can form different hole structures. If all holes connect to each other in series, it is called a ladder hole, such as the cases in Figures 1(c) and (d), which contain two and three holes connected in series, respectively. If the hole breaks through the surface of the part it is called a “through hole”, otherwise it is called a “blind hole”. Holes in a ladder hole generally can be simplified simultaneously. If several holes inside a larger hole are arranged in parallel, such as the case in Figure 1(e), then it is called a parallel hole structure. Hole simplification in this structure should be considered case by case. Related holes may be across different mold parts, such as the case in Figure 1(f), each of which could be either a single hole or a ladder hole on CAD models.

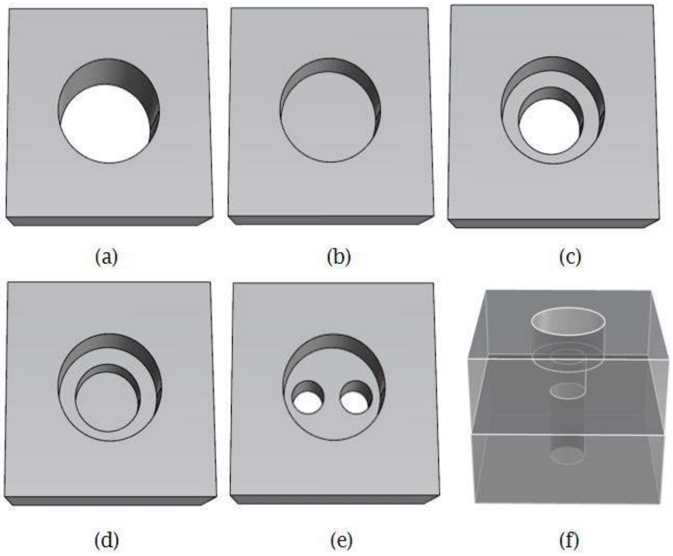

Figure 1: Classification of hole features for feature recognition and simplification, (a) single-through hole, (b) single-blind hole, (c) ladder-through hole, (d) ladder-blind hole, and (e) parallel hole structure, and (f) related holes across different components.

A hole is essentially composed of three types of faces, namely base, side, and bottom faces. A base face is a face where a hole resides; the hole can form an inner loop on the base face. A bottom face is at the bottom of a hole. The side face is a face that connects to both base and bottom faces simultaneously. The angle between the side and base faces is always convex. A blind hole has only one loop and the angle between the side and bottom faces is concave. The base and bottom faces of a blind hole can easily be distinguished. A through hole have two loops and the angles of the faces on both loops are convex. As both angles are convex, the face that is adjacent to the input loop is regarded as the base face, while the face that is adjacent to the second loop is regarded as the bottom face. A fillet may exist between the base and side faces, or the side and bottom faces. The topological data between a fillet and its neighboring faces can be obtained from the database of fillets.

The proposed hole recognition and simplification method uses a set of B-rep models for input, representing the entire set of an injection mold. Although the B-rep models are assembled properly when they are input to the CAD system, they are arbitrarily recorded in the data structure. The output of the proposed method is to recognize and correlate each set of related holes across different CAD models. Those that will be used in the mold flow analysis are preserved, while the others are automatically simplified. The proposed hole recognition and simplification method is divided into four main phases: (1) hole recognition on each CAD model, including single and ladder holes, (2) recognition of related holes across different CAD models, (3) detection of holes that should be preserved, and (4) simplification of unnecessary holes. A detailed description for each of these steps is given below.

3 HOLE RECOGNITION ON EACH CAD MODEL

Before hole recognition, edge and face AAG databases must be computed, which primarily record the geometric and topological data of each edge and face on the CAD models. In particular, the composition of edges on each loop, where a loop can reside either on a single surface or across faces that are G^0, G^1, or G^2 continuous, is recorded as loop data. In addition, fillet data is recorded for the corresponding blend and neighboring faces for each fillet. A detailed description of edge and face AAG databases is given in [6].

Holes on each CAD model include single and ladder holes. The composition of the faces for each hole is evaluated first, and then holes that connect to each other in series are evaluated and formed as ladder holes. The procedure for this algorithm is shown in Figure 2, in which three types of holes, with different end faces and hole structures, are illustrated. The inputs are the loop and fillet data on each CAD model and the outputs are the composition of the faces for each hole and the topological data of the ladder holes. Figure 2(a) depicts the input loop data. This algorithm can be divided into four steps.

First, the facial composition is determined for each hole. The hole search is performed loop by loop. For each loop, the base face that it resides on is found. If all edges on this loop are convex, then a hole exists and all the neighboring faces are recorded. If the neighboring face is not a fillet, then it is considered as a side face. In contrast, if it is a fillet, then the neighboring face of this fillet is considered as a side face. Once all side faces are obtained, their neighboring faces can be found. Similarly, if the neighboring face is not a fillet, then it is regarded as a bottom face. In contrast, if it is a fillet, then the neighboring face of this fillet is regarded as a bottom face. Figure 2(b) shows that four holes corresponding to the case in Figure 2(a) are detected, where all side and bottom faces are colored in yellow.

Second, each hole is evaluated as either through or blind. Once all holes are obtained, each hole is checked, one by one, to attribute it as either blind or through. If all edges between the side and bottom faces are concave, then it is a blind hole. If all edges between the side and bottom faces are convex, then it is a through hole. It should be noted that a through hole is counted twice, as the loops on both of its end faces are checked individually. Therefore, repeated holes should be detected and removed. Figure 2(c) shows that two blind holes and two through holes are detected, where the blind and through holes are colored in red and pink, respectively.

Third, related holes on each CAD model are detected. The relationship between two neighboring holes is obtained by recording the indices of faces that are shared bilaterally. Several holes that connect to each other in series can be regarded as a ladder hole. The cross section of a ladder hole is similar to that of a ladder, which is why it is termed a ladder hole. A ladder hole is composed of a series of holes connected in sequence, with each of the holes called a layer; the first hole is the first layer, the second hole is the second layer, etc. For a hole to be the first layer, the following four conditions should be checked:

(1) Is it a blind hole?
(2) Does its bottom face connect to another hole?
(3) Does its base face not connect to any hole?
(4) Does its base face connect to another hole, which itself connects to multiple holes on the bottom face?

If conditions (1), (2), and (3), or (1), (2), and (4) are satisfied simultaneously, then this hole is regarded as the first layer of a ladder hole. Once the first layer is obtained, the second layer can be obtained by searching for the holes connected to the bottom face of the first layer. Such a search is repeated continuously until the final layer is obtained. The limitation of the proposed approach is that a hole must have at least one loop. If a hole intersects with other features and no loop is formed, then such a hole cannot be recognized.

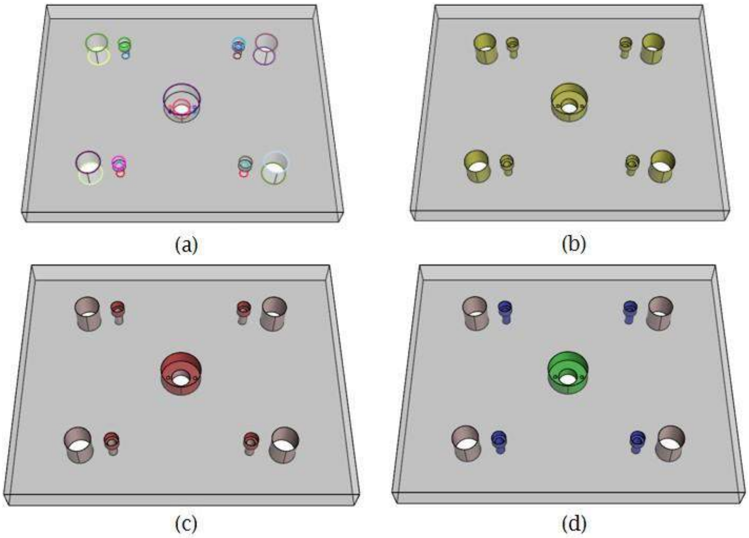

Figure 2: Hole recognition procedures: (a) input loop data (each color denotes a loop), (b) search face composition (yellow) of each hole, (c) compute type and associated properties of each hole (pink: through hole, red: blind hole), and (d) detect related holes (blue: ladder hole, green: parallel hole structure).

Table 1 lists the recorded hole data. The indices H4 – H5 denote the composition of the faces of a hole, and H8 – H12 denote the ladder hole data of a hole, where H8 denotes the index of the ladder hole, H9 denotes the layer of this hole in the ladder hole, H10 denotes the hole connected to the base face of this hole, H11 denotes the hole that connects to the bottom face of this hole, and H12 denotes other holes in the same ladder hole.

Code	Attribute		Remark
H1	Hole index		Index of this hole
H2	Shape		Circular or non-circular shape
H3	Loops		Base and bottom loops
H4	Base face index		Indices of base faces
H5	Side face index		Indices of side faces
H6	Bottom face index		Indices of bottom faces
H7	Hole type		Through or blind hole
H8	Ladder hole	Ladder hole index	Index of the ladder hole, 0: does not belong to a ladder hole
H9		Layer	Layer of this hole on the ladder hole
H10		Base face related	Hole connected to the base face of this hole
H11		Bottom face related	Hole connected to the bottom face of this hole
H12		Members	Other members on the same ladder hole

Table 1: Attributes of a hole.

4 RECOGNITION OF RELATED HOLES ACROSS DIFFERENT CAD MODELS

When several CAD models representing different mold plates are input to a CAD system, no topological relationship exists between these CAD models. To detect related holes across different CAD models, it is necessary to establish the relationships between CAD models that are in contact. As Figure 3 depicts, the proposed algorithm is divided into four steps. First, the mold plates are sorted by using the parting direction: the order of the mold plates in the CAD system is determined. Second, the relationships among the plates are generated (Figure 3(a)). All CAD models that are in contact with each other are detected by checking the intersection of their bounding boxes. Third, the contact faces are searched for (Figure 3(b)): the contact faces of two adjacent CAD models can be used to evaluate the holes passing across different CAD models. Finally, the relationship between the holes passing across different plates is generated (Figure 3(c)). For holes passing across different CAD models, each of the holes can generate a loop on the corresponding contact face. By checking the loops on a pair of contact faces from each of the two adjacent CAD models, the holes passing across different CAD models can be detected. A detailed description for each of the proposed steps in the recognition algorithm is given below.

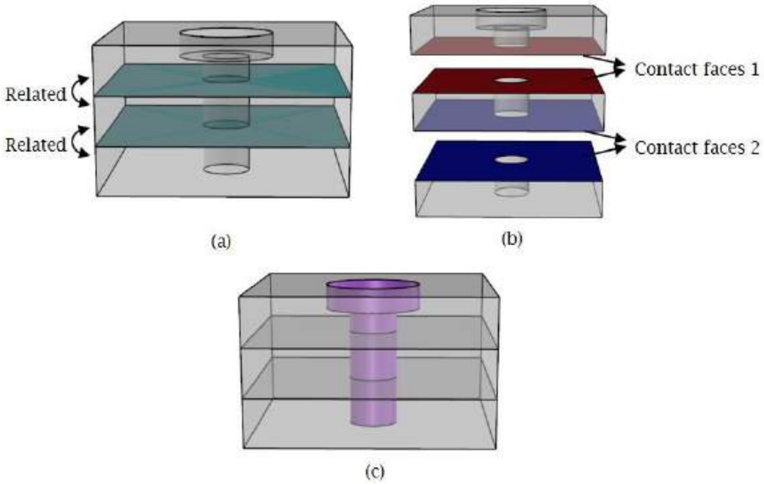

Figure 3: Procedures of detecting related holes across different parts: (a) generate the relationship between parts, (b) search the contact faces, and (c) generate the relationship of holes across different parts.

4.1 Sort the Mold Plate by Using the Parting Direction

When the mold plate CAD models are input to a CAD system, no order exists among these CAD models in the data structure. However, they should be sequentially arranged for subsequent steps. This is done by calculating the order of each mold plate by using the parting direction. The parting direction is a parameter determined by the user, with the default being the positive *z* direction.

4.2 Generate the Relationship among Plates

A simple and computationally efficient method, which employs the bounding boxes of all CAD models, was developed to evaluate adjacent CAD models. Figure 4(a) depicts the concept of checking the intersection of two CAD models by using their bounding boxes. The bounding box of a CAD model is represented by two marginal points, $P_{min} = (x_{min}, y_{min}, z_{min})^T$ and $P_{max} = (x_{max}, y_{max}, z_{max})^T$. An algorithm that compares the *x*, *y,* and *z* coordinates of two sets of marginal points was developed to check the intersection of two bounding boxes. If two CAD models intersect, then the region of intersection can be represented as a range box, as shown in Figure 4(a), which is essentially another bounding box. In contrast, if two CAD models do not intersect, then no range box is found. If two CAD models are in contact at a plane, then the range box is reduced to a plane, as shown in Figure 4(b), where the minimum and maximum coordinate values along the *x*, *y,* or *z* axis are equal.

In the proposed procedure, the bounding boxes of all CAD models are checked pair by pair. A range box is obtained when two CAD models are in contact with each other. Each range box is represented by two marginal points. The CAD models corresponding to each range box are recorded too.

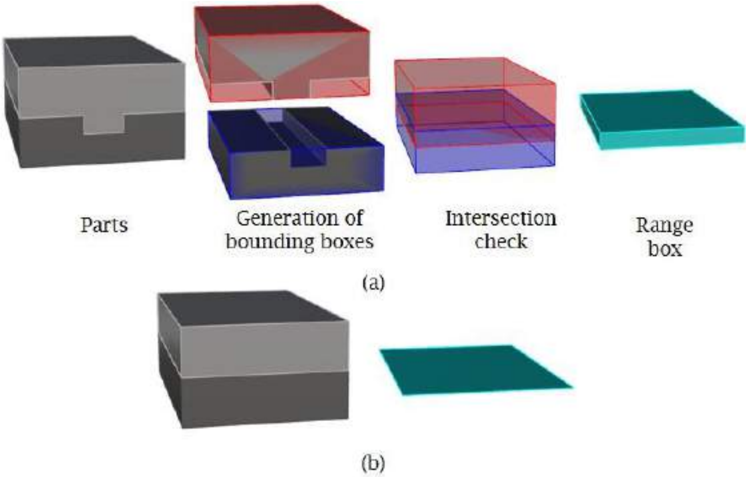

Figure 4: Determine the range box of two intersected CAD models, (a) the concept of evaluating the range box of two CAD modes, and (b) the range box is reduced to a plane when two CAD models contact at a plane.

4.3 Search for Contact Faces

The range box of two adjacent CAD models is then employed to search for contact faces. First, a set of candidate contact faces from each CAD model is evaluated. The bounding box of each face on a CAD model is compared with the range box. If they intersect, then the corresponding face is inside the range box. Two sets of candidate contact faces will be obtained, one from each CAD model. Second, the faces on both sets of candidate contact faces are compared one by one to find the contact faces. Figure 5 depicts the contact conditions of two faces at the contact region. The computational procedure depends on the type of candidate contact faces.

(1) Both candidate contact faces are planes: The surface normal of these two planes is compared. If they are parallel to each other and point in opposite directions, then the intersection of these two planes is checked. If they intersect, then these two candidate planes are regarded as a pair of contact faces. Otherwise, they are not regarded as a pair of contact faces. When they belong to a pair of contact faces, a closed intersection profile is computed and the contact condition is judged in accordance with this profile. Figures 5(a) –(c) belong to this type of plane-to-plane contact. Both contact planes are completely in contact in Figures 5(a) and (b), but with equal and unequal size on the contact planes, respectively. In Figure 5(c), both contact planes are partially in contact.
(2) Both candidate contact faces are surfaces: As the surface normal of both surfaces are not constant, an intersection check of these two surfaces must be implemented. If they intersect, then these two surfaces are considered a pair of contact faces (see Figure 5(d)). Otherwise, they are not regarded as a pair of contact faces. It should be noted that the intersection check of two surfaces is in general, a time consuming operation.
(3) One is a plane and the other is a surface: As a plane and a surface are not in contact with each other, this pair of faces is not regarded as a pair of contact faces.

In the proposed procedure, the faces on both sets of candidate faces are checked pair by pair. When a pair of contact faces is obtained, they are put into a stack. The CAD models corresponding to each pair of contact faces are recorded too.

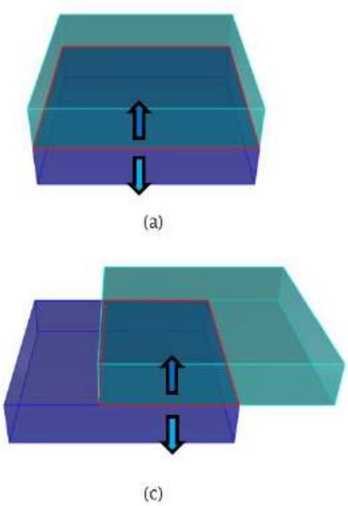

Figure 5: The contact conditions of two CAD models, (a) two equal faces completely contact, (b) two unequal faces completely contact, (c) two faces partially contact, and (d) the contact faces are surfaces.

4.4 Generate the Relationship between Holes across Different Plates

At this stage, holes on each CAD model have already been recognized and the ladder holes have been grouped. The main issue here is to generate the relationship between holes on each pair of contact faces. All groups of holes passing across different CAD models can thus be integrated and arranged in sequence. The proposed process is divided into two main steps. First, the holes on the contact faces are located. Each contact face is essentially a base face or bottom face of a hole. Therefore, the holes that are located on each contact face can be obtained by searching the hole dataset. After the first step, the two sets of holes corresponding to a pair of contact faces are obtained. Each set of holes is located on one of the contact faces. Second, coaxial holes are searched for across different CAD models and grouped together. For the two sets of holes corresponding to the same pair of contact faces, coaxial holes are regarded as a group. Each of the holes on one contact face is compared with those on the other contact face. If two holes are coaxial, then the following three conditions are checked:

(1) Two holes are not recorded in any group: these two holes are put into a new group.
(2) One of the holes is already recorded in a group: when a hole has been recorded in a group, the other hole is placed into the same group.
(3) Both holes are already recorded in different groups: these two groups are combined into one group.

The above process is implemented for all pairs of contact faces. The example shown in Figure 6(a), where two pairs of contact faces are located on three parts (P1~P3), is employed to illustrate how a group of coaxial holes from different CAD models are sequentially integrated. The procedure for integrating all the holes for this example was as follows:

(1) From the first pair of contact faces, two sets of holes were obtained (Figure 6(b)). One set of holes was obtained from the contact face located on Part 1 (red) and the other set was obtained from the contact face located on Part 2 (blue).
(2) The coaxial holes were determined from the two sets of holes. Two new groups resulted from this step (Figure 6(c)).
(3) The ladder hole data was used to extend the groups (Figure 6(d)).
(4) From the second pair of contact faces, two sets of holes were obtained (Figure 6(e)). It should be noted that the holes located on Part 2 are already recorded in the groups.
(5) The holes located on Part 3 were added into group 1 and group 2, respectively.

In the proposed procedure, each hole is checked one by one. If it is not located on any contact face, then this process does not need to be implemented. However, if it is located on a contact face, then all pairs of contact faces with holes with the same coaxial axis are evaluated. Then, the above procedure is implemented to group all the coaxial holes from the different CAD models. Whenever a hole is checked or used, it is marked. Such a process is implemented continuously until all holes have been marked.

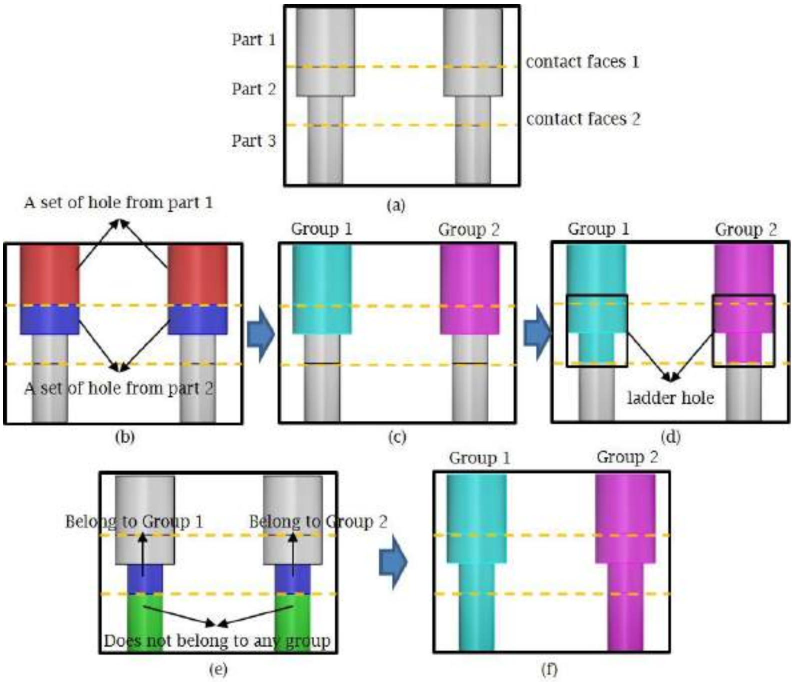

Figure 6: The procedure for integrating the holes across different plates.

5 DETECTION OF HOLES THAT SHOULD BE PRESERVED

In mold flow analyses, holes on the core, cavity, runner system, and cooling channels should be preserved, as they are included in the mesh model. Holes on the core, cavity, and runner system can easily be preserved as they belong to part of the product. However, cooling channels are embedded inside the mold, and cannot be recognized by using the above-mentioned hole recognition method. To solve this problem, the following geometric properties of cooling channels should be considered. Firstly, a cooling channel has a circular cross section, and is exposed to the part surface at its entry and exit, as shown in Figure 7(a). Secondly, a cooling channel can either be a straight tube passing through a mold plate, or the intersection of multiple tubes passing through one or several mold plates, as shown in Figure 7(b). If it is a straight tube, then the tube can be recognized by using the above-mentioned hole recognition method. However, if it contains multiple tubes that intersect each other, only the tubes exposed to the part surface can be recognized, whereas the others embedded inside the mold cannot be recognized, as shown in Figure 7(c). The recognition of cooling channels is divided into three steps: (1) computing the intersected tubes, (2) computing the side faces of the mold, and (3) searching for cooling channels.

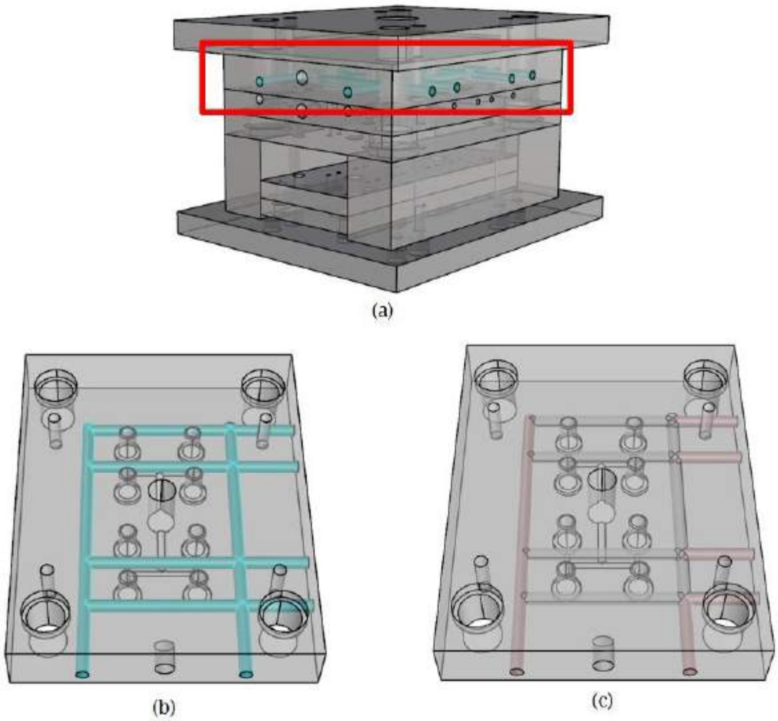

Figure 7: Geometric properties of cooling channels on a mold, (a) entries and exits are located on side faces of the mold, (b) contain multiple tubes intersected each other, and (c) not all tubes are exposed to the part surface.

5.1 Computing Intersected Tubes

The cylindrical face of a tube can either be composed of one or two faces, as shown in Figure 8(a). Therefore, cylindrical faces of the same tube should be grouped first. For all cylindrical faces, three rules are tested: (1) two cylindrical faces must be adjacent to each other through their linear edges, (2) for both cylindrical faces, each surface normal must be the inverse of its radial direction, and (3) the cross sectional edges of two cylindrical faces must be able to form a closed circle. If all three rules are satisfied, then these two cylindrical faces belong to the same tube, and hence are grouped.

Figure 8(b) shows the result of grouping cylindrical faces for the cooling channels presented in Figure 7(b).

The procedure to compute the intersected tubes can then be performed. Starting from a recognized through hole, its bottom face is checked if it belongs to a group of cylindrical faces. If it does not, then this hole is skipped and the next through hole is selected. If it does, then the neighboring group of cylindrical faces is searched for, starting from this group of cylindrical faces. This search is continued until no more neighboring groups of cylindrical faces are found. Figure 8(c) depicts the result of this process. All neighboring cylindrical faces obtained in this step are recorded as a set of intersected tubes.

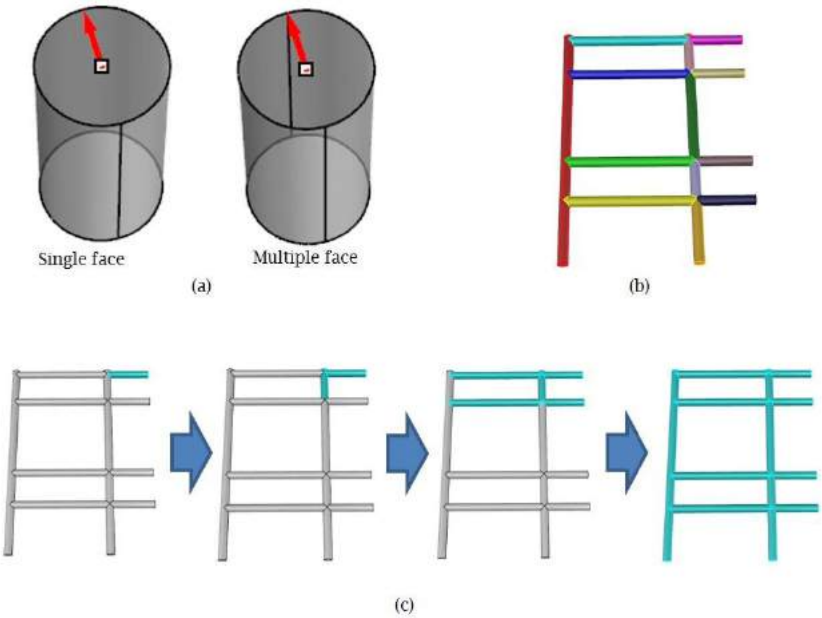

Figure 8: Computing intersected tubes, (a) two kinds of the face composition of a tube, (b) result of grouping cylindrical faces, and (c) the procedure to compute the intersected tubes.

5.2 Computing Side Faces of the Mold

The side faces of a mold are evaluated by using the contact faces obtained previously. For each contact face, its outer loop is employed to search for adjacent faces, as shown in Figure 9(a). The surface normal is checked to see if each adjacent face is the same as that of the bounding plane. If it is the same, then this adjacent face is a side face. Figure 9(b) shows one of the mold parts, where the red color denotes the contact faces and the orange color denotes the adjacent faces. Figure 9(c) denotes the final side faces obtained for the mold.

5.3 Searching for Cooling Channels

As the entries and exits of a cooling channel are located on the side faces of a mold, any entry or exit hole can be found and regarded as a seed hole to search for the entire cooling channel. While not all holes on the side faces belong to cooling channels, entry or exit holes can be distinguished by their property of being both a through hole and an intersected hole. In this step, the base faces of all holes are checked. If the base face of a hole is located on the side face of the mold, then it indicates that this hole is the entry or exit of the cooling channel. Then, the entire cooling channel can be obtained simply by searching for the neighboring faces, one by one.

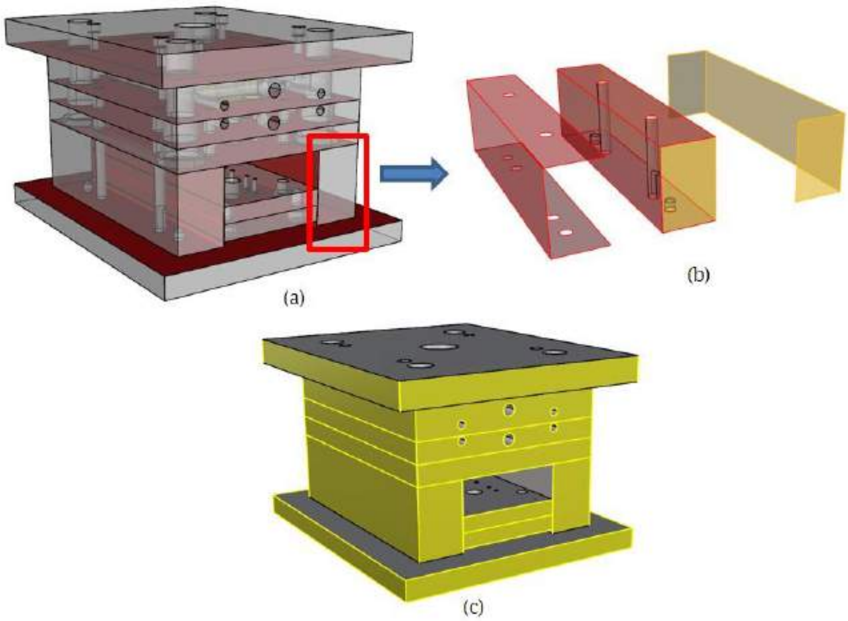

Figure 9: Computing side faces of the mold, (a) contact faces between different mold plates, (b) contact faces and side faces of one of the mold part, (c) final side faces obtained for the mold.

Figure 10 is employed to illustrate the procedures of searching for cooling channels across different plates. In Figure 10(a), Part1 denotes a mold plate where the entry and exit of a cooling channel locates, and Part2 denotes a core or cavity plate where the cooling channel passes through. Figure 10(b) shows the transition of Part1 and Part2, where two through holes H_1 and H_2 are located on Part1 and one intersected tube T_c is located on Part2. When the relationships of holes across different plates are generated, H_1 and T_c will be recorded as a group first. Then, the hole H_2, which is adjacent to the other side of T_c, is searched and grouped. Finally, H_1, H_2 and T_c are grouped together. This group has two loops on the side face of the mold base, so all holes in this group are recorded as a cooling channel. When all holes on the side faces of all mold plates are checked, all cooling channels can thus be obtained.

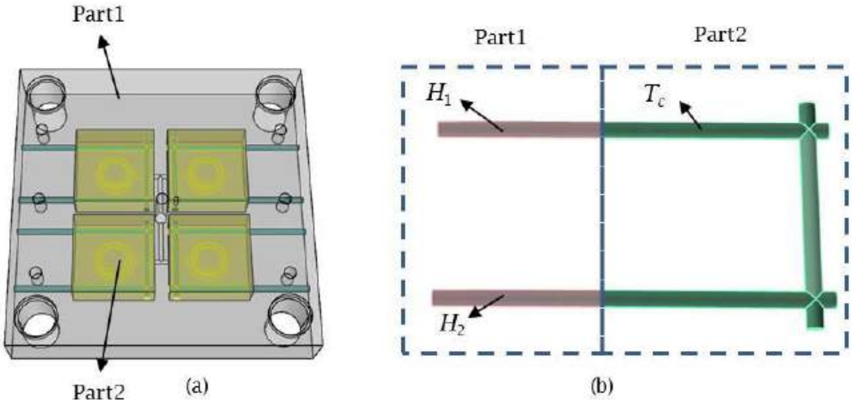

Figure 10: The cooling channels across different plates.

6 HOLE SI MPLIFICATION

Most holes in the CAD models of an injection mold can be simplified in mold flow analyses, except in two situations. When a hole (both circular and non-circular types) is too large it is considered as part of the designed shape and should be preserved. A maximum hole perimeter is assigned in the proposed hole simplification algorithm to filter out such holes. Additionally, holes on the core, cavity, runner system, and cooling channels must be preserved in the analysis. These parts can be excluded manually while the hole simplification is implemented.

Hole simplification in a B-rep model involves essentially changing the faces and associated entities related to the hole to remove the hole itself. As a hole is essentially composed of three kinds of faces, the base, side, and bottom faces, when a hole is removed, some of its faces are removed from the data structure, whereas the others are modified. Take the mold plate shown in Figure 11(a) as an example, which includes three kinds of holes: runner, through and ladder-through holes. The runner hole is preserved. For a through hole, as shown in Figure 11(b), the base faces on both sides are modified by removing their inner loops, whereas the side face is removed. For a ladder-through hole, as shown in Figure 11(c), the base and bottom faces of the first and last layers, respectively, are modified, whereas all other faces are removed. For the through hole and the ladder-through hole, the associated loops, edges, trims, and vertices in the B-rep data are either deleted or modified too. The sample model after the hole simplification was performed is shown in Figure 11(d).

Holes are divided into two types, single or ladder, and then further be subdivided into either blind or through. The faces that should be modified or removed for each type are (see Figure 12):

(1) For single-through holes (Figure 12(a)), the base and bottom faces are modified, while the side face is removed.
(2) For single-blind holes (Figure 12(b)), the base face is modified, while the side and bottom faces are removed.
(3) For ladder-through holes (Figure 12(c)), the base and bottom faces of the first and last layers, respectively, are modified, while all other faces are removed.
(4) For ladder-blind holes (Figure 12(d)), the base face of the first layer is modified, while all other faces are removed.

The operations required for the modification and removal of each face are similar to the case shown in Figure 11. That is, when a face is modified, its inner loop and associated entities are deleted from the B-rep data. In addition, when a face is removed, the face and associated entities are deleted from the B-rep data. For cases (1) and (2), each of the single holes is individually simplified. However, for cases (3) and (4), all holes of a ladder hole can be removed either completely or partially. For partial removal, the lower layers of holes should be removed while the upper layers of holes are preserved. Such an operation can be done manually.

For the parallel hole structure, as shown Figure 1(e), it is possible to select one hole in the hole structure, and then use the common faces on both sides to find its neighboring holes, sequentially. Once all related holes are found, all faces that should be removed or modified can be determined based on the type of faces on the base and bottom; the judgment is similar to the one used for ladder holes.

Once the faces to be removed and modified are determined, the removal of the faces is performed first. The topological and geometric data structure in the B-rep model should also be modified simultaneously. That is, when a face is removed, all loops, edges, trims, and vertices related to this face should be removed at the same time. When some edges and vertices are shared by other faces that are not removed, these edges and vertices should be kept. For example, if an edge is adjacent to both base and side faces, where the former is to be modified while the latter is to be removed, then this edge and associated vertices should be kept when the side faces and their associated entities are removed. The face modifying operation should be implemented after the removal of the faces and their associated entities.

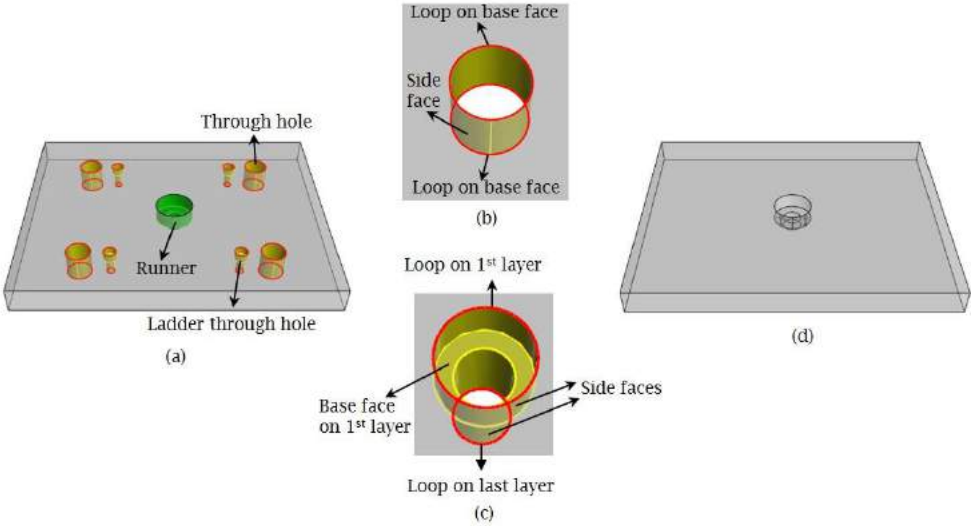

Figure 11: Example of hole simplification: (a) a mold plate before hole simplification, (b) faces to manipulate for a through hole, (c) faces to manipulate for a ladder-through hole, and (d) the mold plate after hole simplification.

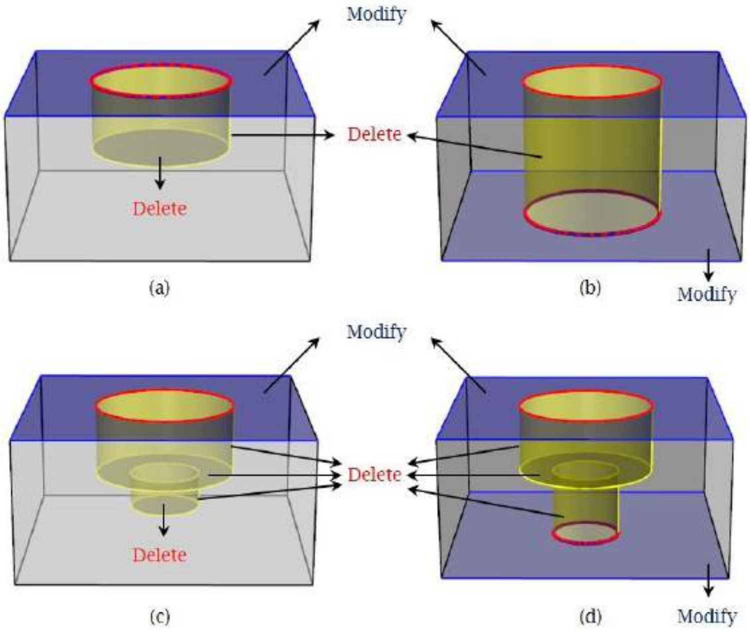

Figure 12: The simplification data for each type of hole: (a) single-blind hole, (b) single-through hole, (c) ladder-blind hole, and (b) ladder-through hole.

7 RESULTS AND DISCUSSION

We tested the feasibility of the proposed algorithms with a program written in C++ and based on the Rhino CAD platform [13] and the openNURBS functions [12]. The proposed algorithm is only compatible with CAD models with manifold topology and geometry. A maximum hole perimeter d_{max}

should be assigned to limit the maximum perimeter of a hole that can be simplified. If the perimeter of a hole (circular or non-circular type) is larger than d_{max}, then it is usually a pocket to be assembled with other mold parts, and hence should be preserved. The default value of 300 mm for d_{max} was used in this study.

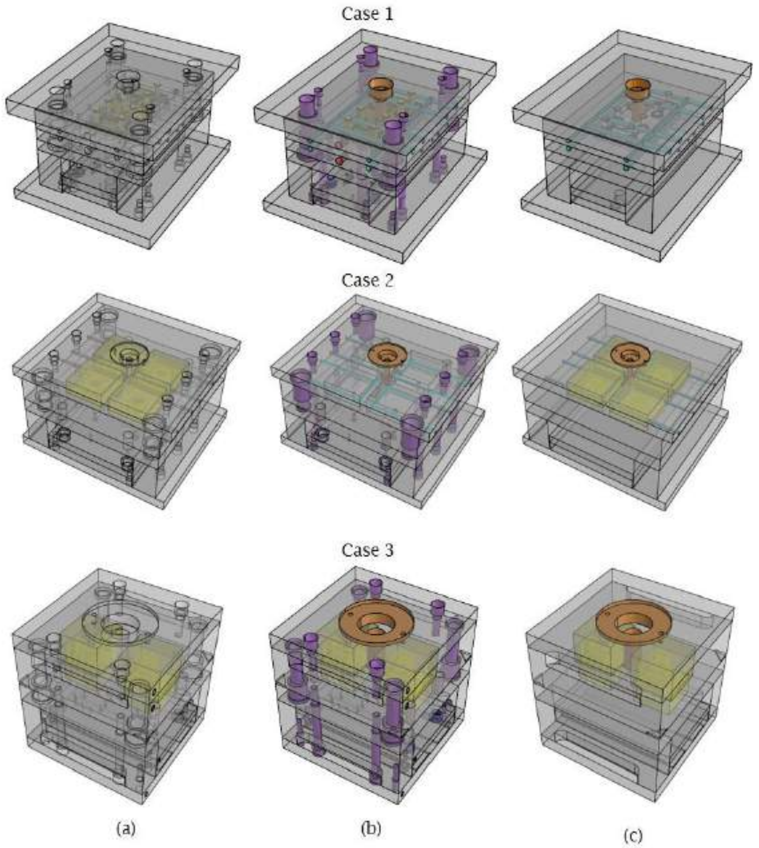

Figure 13: Three injection molds used in this study, (a) the original CAD models, (b) hole recognition, and (c) hole simplification.

Figure 13 illustrates the hole recognition and simplification results for three injection molds, where Figures 13(a), (b), and (c) denote the original CAD models, hole recognition, and hole simplification, respectively. Each example contains all CAD models of a mold set, including many different kinds of mold plates, both core and cavity. Most of the holes are exposed to part surfaces, except for the cooling channels, which are embedded inside the mold plates. The holes recognized on the mold plates are classified as the runner, cooling channels, single-blind, single-through, ladder-blind, or ladder-through. The runner and cooling channels are in green and water blue, respectively, while each type of the other holes are also in different colors, as shown in Figure 13(b). Only the sprue and cooling channels were preserved, while all other holes on the mold plates were removed, as shown in Figure 13(c).

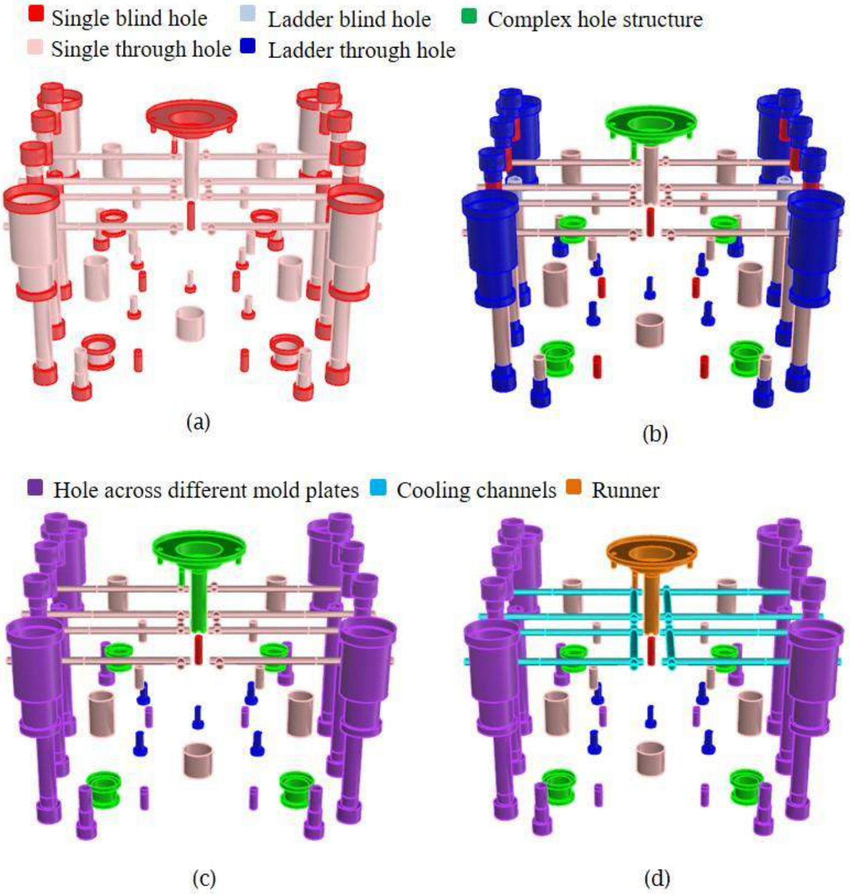

Figure 14: Immediate results of hole recognition for Case 2, (a) holes are initially divided into through and blind types, (b) ladder holes and parallel hole structure are recognized, (c) co-axial holes across different CAD models are detected, and (d) the runner and cooling channels are detected.

Figure 14 illustrates the results of the hole recognition performed for Case 2 in Figure 13. All holes were individually recognized by using the loop data first. Through and blind holes can be separated in this step, as shown in Figure 14(a), where pink and red represent through and blind holes, respectively. Next, all adjacent holes on the same CAD model were grouped and formed into different hole structures, including ladder and parallel hole structures, as shown in blue and green, respectively, in Figure 14(b). It should be noted that single-through and single-blind holes are still shown in pink and red, respectively. Coaxial holes across different CAD models were then searched and grouped, as shown in purple (see Figure 14(d)). The other types of holes, namely ladder and parallel hole structures, single-through holes, and single-blind holes are still shown in their respective colors. Finally, holes on the side faces of the mold were used to search for cooling channels, and the hole on the first mold plate is used to evaluate the runner, which are shown in water blue and orange, respectively, in Figure 14(d). In total, there are seven types of holes, namely the runner, cooling channels, ladder holes passing across different CAD models, ladder holes contained within a CAD model, parallel hole structures, and single-through single-blind holes. Each type of hole is individually stored in a data structure.

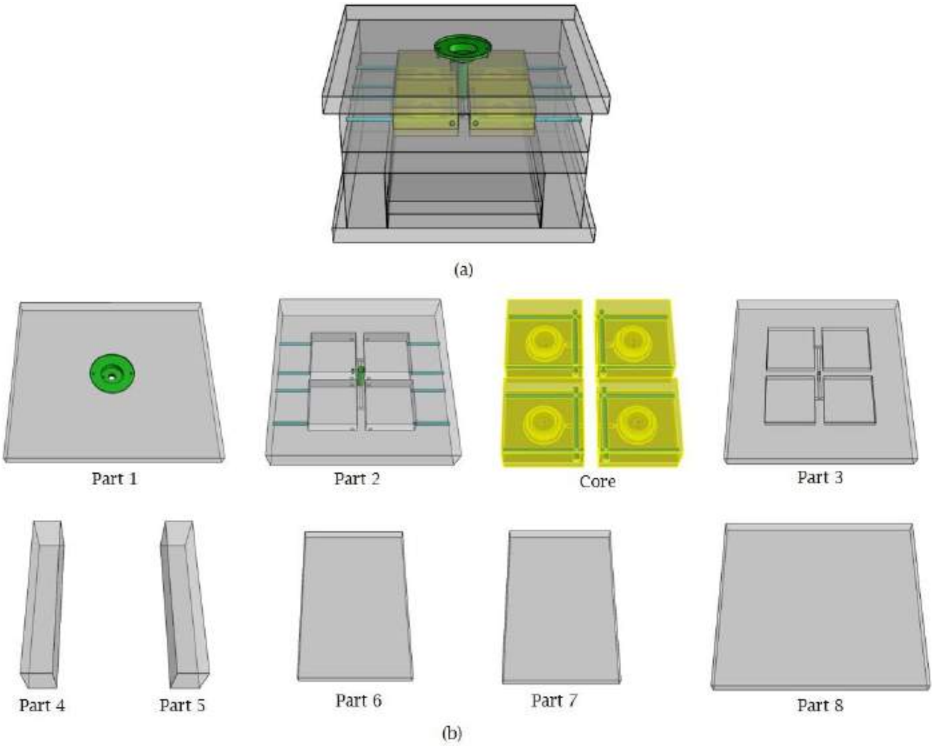

Figure 15: Hole simplification results for the first mold base: (a) assembly of all mold components and (b) each mold component.

Figure 15 illustrates the result of the hole simplification on the mold plates from Case 2 from Figure 13, where Figure 15(a) shows the assembly of the mold set while Figure 15(b) shows each mold plate. It should be noted that only holes related to the runner and cooling channels were preserved, while the other holes were deleted. The green holes on Parts 1 and 2 belong to the runner. The central hole on Part 1 was evaluated first. All holes related to this hole and passing across different mold pates were then found and regarded as the runner. In addition, the cooling channels pass through both Part 1 and the core plate. This indicates that when searching for the cooling channels, holes that cross different CAD models should be detected also. For the other five types of holes, namely ladder holes passing across different CAD models, ladder holes contained within a CAD model, parallel hole structures, and single-through and single-blind holes, each of them can be deleted in accordance with its own structure, as mentioned previously.

Tables 2 to 4 show the results of the proposed method for Cases 1 to 3, respectively. Each table lists the number of holes in each mold plate for the different molds. The types listed include single-blind, single-through, ladder-blind, ladder-through, and complex hole structures. The first four types are as mentioned previously, and the complex hole structure contains parallel hole structures, and intersected tubes or other miscellaneous structures. It can be seen from these tables that most of the holes in injection molds belong to the first four types, and only few of them belong to the complex hole structures, such as runners. Case 1 and Case 2 have some complex hole structures marked with an "*" on the table. These holes are actually cooling channels. However, the recognition procedure of the cooling channels is different from other types of holes, so they are marked on the tables. The CAD models for Case 3 do not contain cooling channels. Only one complex hole structure exists in this case, which is the runner.

Case 1	No. Holes	Time (s)	Recognition results					
			Single		Ladder		Complex Structure	Success / Failure / Misjudgment
			Blind	Through	Blind	Through		
Part 1	9	0.351	0	4	0	4	1	9/0/0
Part 2	18	0.493	13	0	0	4	1*	17+1*/0/0
Part 3	34	0.412	1	30	0	1	2*	32+2*/0/0
Part 4	31	0.354	0	27	0	4	0	31/0/0
Part 5	4	0.274	2	2	0	0	0	4/0/0
Part 6	4	0.293	2	2	0	0	0	4/0/0
Part 7	27	0.197	4	2	0	21	0	27/0/0
Part 8	17	0.115	10	3	0	4	0	17/0/0
Part 9	11	0.308	0	1	0	8	2	11/0/0
Total	155	2.797	32	71	0	46	3+3*	121+3*/0/0

Table 2: Results of Case 1.

Case 2	No. Holes	Time (s)	Recognition results					
			Single		Ladder		Complex Structure	Success / Failure / Misjudgment
			Blind	Through	Blind	Through		
Part 1	7	0.213	0	0	0	6	1	7/0/0
Part 2	16	0.345	7	1	0	4	4*	12+4*/0/0
Part 3	19	0.497	1	8	6	4	0	19/0/0
Part 4	3	0.249	0	3	0	0	0	3/0/0
Part 5	3	0.278	0	3	0	0	0	3/0/0
Part 6	13	0.168	0	4	0	5	4	13/0/0
Part 7	8	0.149	4	0	0	4	0	8/0/0
Part 8	11	0.348	4	1	0	6	0	11/0/0
Total	80	2.247	16	20	6	29	9	76+4*/0/0

Table 3: Results of Case 2.

Case 3	No. Holes	Time (s)	Recognition results					
			Single		Ladder		Complex Structure	Success / Failure / Misjudgment
			Blind	Through	Blind	Through		
Part 1	5	0.204	0	0	0	4	1	5/0/0
Part 2	9	0.243	4	0	0	5	0	9/0/0
Part 3	12	0.239	4	4	0	4	0	12/0/0
Part 4	4	0.111	2	2	0	0	0	4/0/0
Part 5	4	0.163	2	2	0	0	0	4/0/0
Part 6	28	0.229	0	4	0	24	0	28/0/0
Part 7	4	0.142	0	0	0	4	0	4/0/0
Part 8	8	0.177	0	4	0	4	0	8/0/0
Total	74	1.508	12	16	0	45	1	74/0/0

Table 4: Results of Case 3.

The CPU time required for Cases 1 to 3 is also shown in Tables 2-4, respectively. Each table lists the CPU time required on each mold plate and the total time required for all mold plates on a mold. The results indicate that the maximum CPU time required for the recognition of all holes on a CAD model is less than 1 sec. The maximum CPU time required to generate the relationship between holes across different plates is less than 0.3 sec. Also, the total CPU time required for all mold plates on a mold is less than 3 sec. Therefore, the computational efficiency of the proposed hole recognition algorithm is quite good. The simulations were performed on a personal computer with an Intel Core i5-4460 CPU 3.2GHz and 12 GB of RAM.

To further evaluate the advantages of the proposed hole recognition algorithm, the proposed method was compared with CADdoctor [2]. Two separate modules are provided in CADdoctor for circular and non-circular holes, respectively; whereas they are recognized simultaneously in the proposed method. The most significant problem found in CADdoctor is that it is difficult to identify all circular and non-circular holes perfectly as both modules have their limitation. When both modules are implemented in sequence, repeated recognition may occur. For the proposed method, however, such a problem would not be happened as both circular and non-circular holes are clearly defined and recognized simultaneously in an algorithm. For simplification, there are two ways to simplify holes in CADdoctor: (1) simplify all holes on, and (2) simplify one hole selected by the user. When a series of mold plates related to a mold need to be handled, CADdoctor must deal with each mold plate one by one. But the proposed algorithm can simplify all unnecessary holes on a mold set automatically. Therefore, the proposed hold simplification algorithm is much more powerful than CADdoctor.

8 CONCLUSION

The development of hole recognition and simplification algorithms for injection molds was studied, and the feasibility of the proposed algorithms by using several model sets was verified. An algorithm for the recognition of all holes on the mold set was proposed. After recognition, it divided all holes into two groups, either the through hole or the blind hole types. Related holes were divided into either the ladder hole or parallel hole structure, with the hole type determined for each hole within all the CAD models. Coaxial holes passing across different CAD models were then evaluated. Finally, because they should be preserved, holes related to the runner and cooling channels were individually detected. A hole simplification algorithm was also proposed. Holes that should be deleted were divided into five types: ladder holes passing across different CAD models, ladder holes contained within a CAD model, parallel hole structures, single-through holes, and single-blind holes. For each hole type, the faces that should be deleted or modified, and their associated entities that should be processed, were analyzed. The operations required for the hole simplification were also addressed. The proposed method can preserve the runner and cooling channels, while deleting all other holes in the mold set. The main contribution of the proposed method for the deletion of unnecessary holes in the mold plates of a mold set is that it is based on feature recognition and feature simplification. In cases where it is necessary to consider the original, non-simplified mold plates of a mold base (typically modeled as a rectangular box for simplification) in a mold flow analysis, the total amount of mesh nodes can be substantially reduced.

9 ORCID

Jiing-Yih Lai, http://orcid.org/0000-0002-0495-0826
Pei-Pu Song, http://orcid.org/0000-0001-5937-5841
Chia-Hsiang Hsu, http://orcid.org/0000-0001-5763-4766
Yao-Chen Tsai, http://orcid.org/0000-0002-3408-8835

REFERENCES

[1] Boussuge, F.; Leon, J.-C.; Hahmann, S.; Fine, L.: Idealized models for FEA derived from generative modeling processes based on extrusion primitives, Engineering with Computers, 31(3), 2015, 513-527. http://dx.doi.org/10.1007/s00366-014-0382-x

[2] CADdoctor, http://elysiuminc.com/products/caddoctor/, accessed on 06-05-2019.

[3] Chong, C.-S.; Kuma, A.-S.; Lee, K.-H.: Automatic solid decomposition and reduction for non-manifold geometric model generation, Computer-Aided Design, 36(13), 2004, 1357-1369. http://dx.doi.org/10.1016/j.cad.2004.02.005

[4] Cui, X.; Gao, S.: Zhou, G.: An efficient algorithm for recognizing and suppressing blend features, Computer-Aided Design and Applications, 1(1-4), 2004, 421-428. http://dx.doi.org/10.1080/16864360.2004.10738284

[5] Juttler, B.; Kapl, M.; Nguyen, D.-M.; Pan, Q.; Pauley, M.: Isogeometric segmentation: The case of contractible solids without non-convex edges, Computer-Aided Design, 57, 2014, 74-90. http://dx.doi.org/10.1016/j.cad.2014.07.005

[6] Lai, J.-Y.; Wang, M.-H.; Song, P.-P.; Hsu, C.-H.; Tsai, Y.-C.: Recognition and decomposition of rib features in thin-shell plastic parts for finite element analysis", Computer-Aided Design and Applications, 15(2), 2018, 264-279. https://doi.org/10.1080/16864360.2017.1375678

[7] Li, B.; Liu J.: Detail feature recognition and decomposition in solid model, Computer Aided Design, 34(5), 2002, 405-414. http://dx.doi.org/10.1016/S0010-4485(01)00118-X

[8] Li, J.; Sun, L.; Peng, J.; Du, J.; Fan, L.: Automatic small depression feature recognition from solid B-rep models for meshing, 2011 International Conference on Electrical and Control Engineering, 2011, 4386-4389. http://dx.doi.org/10.1109/ICECENG.2011.6057432

[9] Li, J.; Tong, G.; Shi, D.; Geng, M.; Zhu, H.; Hagiwara, I.: Automatic small blend recognition from B-rep models for analysis, Engineering with Computers, 25(3), 2009, 279-285. https://doi.org/10.1007/s00366-009-0127-4

[10] Li, Y.-G.; Ding, Y.-F.; Mou, W.-P.; Guo, H.: Feature recognition technology for aircraft structural parts based on a holistic attribute adjacency graph, Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, 224(2), 2010, 271-278. http://dx.doi.org/10.1243/09544054JEM1634

[11] Makem, J.-E.; Armstrong, C.-G.; Robinson, T.-T.: Automatic decomposition and efficient semi-structured meshing of complex solids, Engineering with Computers, 30(3), 2014, 345-361. http://dx.doi.org/10.1007/s00366-012-0302-x

[12] openNURBS, http://www.rhino3d.com/tw/opennurbs, accessed on 06-05-2019.

[13] Rhinoceros, http://www.rhino3d.com, accessed on 06-05-2019.

[14] Song, P.-P.; Lai, J.-Y.; Tsai, Y.-C.; Hsu, C.-H.: Automatic recognition and suppression of holes on mold bases for finite element applications, Engineering with Computers, 2018, 1-20. https://doi.org/10.1007/s00366-018-0640-4

[15] Sundararajan, V.; Wright, P.-K.: Volumetric feature recognition for machining components with freeform surfaces, Computer Aided Design, 36(1), 2004, 11-25. https://doi.org/10.1016/S0010-4485(03)00065-4

[16] Sunil, V.-B.; Pande, S.-S.: Automatic recognition of features from freeform surface CAD models, Computer Aided Design, 40(4), 2008, 502-517. https://doi.org/10.1016/j.cad.2008.01.006

[17] Venkataraman, S.; Sohoni, M: Blend recognition algorithm and applications, The sixth ACM Symposium on Solid Modeling and Applications, 2001, 99-108. http://dx.doi.org/10.1145/376957.376970

[18] You, C.-F.; Tsai, Y.-L.; Liu, K.-Y.: Representation and similarity assessment in case-based process planning and die design for manufacturing automotive panels, International Journal of Advanced Technology, 51(1-4), 2010, 297-310. https://doi.org/10.1007/s00170-010-2609-3

[19] Zhang, C.-J.; Zhou. X.-H.; Li, C.-X.: Automatic recognition of intersecting features of freeform sheet metal parts, Journal of Zhejiang University-Science A, 10(10), 2009, 1439-1449. https://doi.org/10.1631/jzus.A0820705