COMPUTER-AIDED DESIGN & APPLICATIONS, 2018
VOL. 15, NO. 2, 264–279
https://doi.org/10.1080/16864360.2017.1375678

Recognition and decomposition of rib features in thin-shell plastic parts for finite element analysis

Jiing-Yih Lai [a], Ming-Hsuan Wang [a], Pei-Pu Song [a], Chia-Hsiang Hsu [b] and Yao-Chen Tsai [b]

[a]National Central University, Taoyuan, Taiwan; [b]CoreTech System (Moldex3D) Co., Ltd., Hsinchu, Taiwan

ABSTRACT

Feature recognition has been extensively studied for applications in computer-aided design (CAD), computer-aided manufacturing, and computer-aided process planning. However, its application in finite element analysis (FEA) is limited. The primary motivation for applying feature recognition in FEA is to facilitate high-quality mesh generation, which requires not only identifying the geometric entities of a feature but also decomposing and computing the region data. This study proposes an approach based on feature recognition to recognize ribs from a CAD model and decompose them into regions that can be meshed with hexahedral or prismatic meshes. Ribs in a CAD model are generally connected to form a complex structure. A rib recognition algorithm to recognize all faces belonging to a rib structure is proposed, and a rib decomposition algorithm was also developed to decompose the rib structure into rib segment and transition regions. After both types of region were defined, the data to decompose each of them from the CAD model was computed. Several examples are presented to verify the results of the rib recognition and demonstrate the feasibility of the proposed rib decomposition algorithm.

KEYWORDS

Rib Recognition; Feature Recognition; Rib Decomposition; Meshing; B-rep Model

1. Introduction

Mold flow analysis is commonly used in injection molding to assist in designing injection molds as well as setting the process parameters. In mold flow analysis, it is necessary to convert the computer-aided design (CAD) model into solid meshes so that the solver can perform computations. Traditionally, tetrahedral meshes are used because they are easy to generate automatically, but this type of mesh is of the lowest quality compared with other types (e.g., pyramid, prism, or hexahedron). Hexahedral meshes are highly preferable to tetrahedral meshes for their greater accuracy, convergence, and application specificity. However, hexahedral meshes are inherently trickier to generate because they require the careful decomposition of the CAD model, which is usually complex and challenging even for well-trained CAE engineers. The generation of entirely hexahedral meshes on a CAD model may be difficult, but a hybrid combination of hexahedral and prismatic meshes is possible and can solve the problems of tetrahedral meshes.

Typical methods for generating regular types of meshes, such as hexahedral and prismatic meshes, are mapping and submapping, meshing primitives, and sweeping. These require the decomposition of some recognized patterns from a CAD model and the conversion of each of them into meshes using one of the aforementioned meshing algorithms [18]. Feature recognition has been studied for decades and several feature recognition methods are available on commercial CAD systems. However, these methods are primarily for applications in CAD, computer-aided manufacturing (CAM), and computer-aided process planning (CAPP); they may not be appropriate for application in finite element analysis (FEA) because the data extracted is not suitable for meshing. Although substantial efforts have been made concerning automatic decomposition and meshing, none of them are reliable enough for industrial applications because real CAD models are more complex, variable, and unpredictable. Combining multiple meshing algorithms and mesh types to deal with practical CAD models is necessary.

Ribs are a common feature in CAD design, appearing frequently in many products. In particular, they are often designed on thin-shell plastic parts to enhance the strength of the structure. Ribs are generally designed as a series of interconnected structures distributed across the inner surface of a thin-shell part to provide the required strength. The shape design and distribution of ribs can

CONTACT Jiing-Yih Lai ✉ jylai@ncu.edu.tw; Ming-Hsuan Wang ✉ a7512cs@hotmail.com; Pei-Pu Song ✉ scarlet811025@gmail.com; Chia-Hsiang Hsu ✉ davidhsu@moldex3d.com; Yao-Chen Tsai ✉ cloudtsai@moldex3d.com

affect the quality of the injecting part. One defect often associated with ribs is that some regions on the outer surface of a thin-shell part may be sunken slightly because of shrinkage of the material near regions on the inner surface where several ribs intersect. An accurate simulation of such a phenomenon is necessary to provide a deeper understanding of the physical mechanisms so that its occurrence can be avoided. As the rib structure on thin-shell parts can be highly complex, an improved meshing method is necessary to adjust the size, shape, and mesh type in various regions of the shape. The traditional meshing method based on tetrahedral meshes is clearly not satisfactory.

Topological data structures are the basis of various feature recognition and decomposition methods. Ansaldi et al. [1] proposed a face adjacency graph (FAG) to record the adjacent relationships of faces and edges, where an edge is denoted as a node on the graph and its adjacent faces are denoted as links of the node. Joshi et al. [9] proposed an attributed adjacency graph (AAG) to enhance shape-comparison efficiency by adding an angle between two neighboring faces at the edge attribute. Marefat et al. [21] proposed an extended attribute adjacency graph (EAAG) to deal with primitives that intersect each other. The depressions are represented as cavity graphs, in which the links reflect the concavity of the intersection between two faces, and the node labels reflect the relative orientation of the faces comprising a depression. Lu et al. [19] and Li et al. [16] proposed a hybrid data structure combining topological data structures (e.g., AAG or EAAG) and heuristic rules (e.g., hint-based or rule-based) for feature recognition. They defined seed faces on the model and searched for the target features in accordance with the proposed hybrid data structure.

Most studies in feature recognition relate to CAD, CAM and CAPP. A CAD model generally consists of many types of features; most of them are depressions (e.g., holes or pockets) or protrusions (e.g., bosses, ribs, or extrusions). Lim et al. [17] proposed an algorithm for recognizing depression and protrusion features (DP-features) on freeform solids. This method uses G^1 continuity between edge segments to identify DP-feature boundaries that cross multiple faces and geometries. Li et al. [14] proposed an edge-based approach for recognizing small depression features in mesh generation. Convex inner loops are formed near edges to recognize depressions. Ismail et al. [7–8] proposed a technique called edge boundary classification (EBC) for recognizing simple and interacting cylindrical- and conical-based features in B-rep models. An EBC pattern was formed from three test points on an edge loop, where the first two points were on the edge loop and the third point was their midpoint. A loop-up table in terms of the EBC pattern was provided to detect various feature types. Zhang et al. [29] proposed a region-based method for recognizing protrusion and depression features. Symbolic computation was employed to characterize the curvature properties of the freeform surface, which could help to decompose the surface into regions. A rule-based approach was then employed to recognize protrusion and depression features in accordance with specific geometrical and topological relations. Shah et al. [26] categorized available feature recognition algorithms into six types: topological, heuristic, symbolic, volumetric, process-centric, and hybrid, in accordance with the application area (e.g., geometric design, machining, stress, or analysis), applicability of geometric types (plane and freeform), topological data structure (e.g., AAG, FAAG, or HAAG), feature-searching method, and feature definition.

The feature data extracted from various feature recognition methods are generally not sufficient for solid mesh generation because they are primarily used to describe feature shape. For the feature data to be useful for mesh generation, they should be designed and extracted in accordance with the needs of particular meshing methods. Several approaches are available in the literature to decompose, extract, or simplify solid models and thereby obtain such data for automatic mesh generation. Wang et al. [28] proposed a boss recognition method and the meshing data that can be used for automatic meshing. Chong et al. [5] focused on idealizing finite element models and proposed some operations to allow the user to automatically reduce the dimensionality of geometric models by using a decomposition and reduction method. However, they essentially approximated a three-dimensional problem with two-dimensional meshes, which might not be sufficient accurate. Boussuge et al. [2–3] proposed an approach based on iterative identification and removal of rib primitives for the recognition of thin walls. They provided a graph containing all nontrivial construction trees using generative processes, which is more useful for evaluating variations of idealization. However, they mainly deal with two-dimensional meshes only. Makem et al. [20] generated shape metrics by 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 nonmanifold assembly of long, slender, and complex subregions. Both structured anisotropic and unstructured meshes could then be employed in different regions to obtain an efficient distribution of meshes. However, the models that can be handled are limited. Juttler et al. [10] 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 for repeatedly decomposing the solid into smaller solids until all of them belonged to a certain class of predefined base solids. Each of the base solids could then be meshed by hexahedral meshes. Although the meshes used are hexahedral meshes, the distribution of the meshes are irregular and cannot be controlled. So, the complexity of the models that can be handled is limited. Also, no fillet is allowed on the models.

Regarding rib definition and recognition, Owodunni et al. [24] defined a rib feature as an additive, prismatic machining feature. They employed properties such as the closure of a feature shape along the x, y, and z axes for the classification of 2.5D types. Li et al. [15] defined ribs as a set of constrained and adjacent faces of a part, which are associated with a set of specific rib machining operations. They formulated a set of rules, based on machining knowledge and combined with a holistic attribute adjacency graph, in a feature recognition algorithm that extracts seed faces from a CAD model, identifies individual local rib elements, and further clusters these elements into groups for machining operations. Zhu et al. [30] proposed a semantic mid-surface abstraction method for the thin-walled models based on rib-feature recognition. The model decomposition was conducted based on the identified rib features and the hierarchical semantic information. All discrete features and model regions were finally stitched to yield the final mid-surface model.

2. Problem statement

Many commercial CAD systems already provide CAD feature trees to record the structure of a CAD model, which also includes the structure of many existing features. However, the commonly used CAD data file formats, such as IGES and STEP, do not record such a constructing structure. For a solid model, the boundary representation (B-rep) data structure is commonly used and its data can be rebuilt from the CAD data file. However, the B-rep data does not record the constructing structure of a CAD model. Thus, when a CAE system inputs such a CAD data, all it can access is the topological and geometric information of the CAD model. On the other hand, in CAE analysis, a hybrid combination of different types of solid meshes has been proved to be better than conventional tetrahedral meshes. It requires the recognition of each interesting region and decomposing it from the rest of the model. A feature in the CAD design may not directly represent a region that can be meshed. In fact, most commonly used features are too complex in geometry, and virtual faces may exist across multiple features, so they cannot be meshed directly. We aim to develop an independent feature recognition and decomposition method that can be implemented in a CAE system or used as a pre-processing tool for a better generation of solid meshes.

The purpose of this study is to investigate the recognition and decomposition of ribs on thin shell parts. To improve the quality of solid meshes for ribs, ribs should first be recognized and decomposed; however, because ribs are usually designed as a series of interconnected structures, it is necessary to decompose them into a set of regions and apply a meshing method for each. Therefore, we have developed a hybrid meshing method combining hexahedral and prismatic meshes for all ribs, which operates on the following principles. First, to determine the decomposition strategy, a rib structure must be divided into regions so that each one can be meshed using either hexahedra or prisms. The decomposition strategy is critical as it affects the feasibility of applying hexahedral and prismatic meshes for all rib regions. To apply hexahedral meshes, a region should have a pair of symmetric or almost symmetric matching faces. At the transition of distinct rib regions, where the shapes are more complex, both hexahedral and prismatic meshes could be applied. If a pair of matching faces can be found on a transition region, it can be meshed with the hexahedral type. If not, the prismatic type is employed. However, it requires an analysis of the shape construction on transition regions and proposing a meshing method for each type. In this study, transition regions are primarily meshed with the prismatic type. Second, a rib recognition algorithm is required to detect all faces corresponding to each of the rib segments (i.e., each region with a pair of matching faces). Third, the data for all regions in a rib structure must be analyzed and computed. Other than the rib segments that have symmetric or almost symmetric matching faces, the transition of varying rib segments in a rib structure must also be recognized and decomposed. Finally, the transition faces between the ribs and the other parts of the model must be evaluated. The transition faces are critical when the entire CAD model is meshed; they can be used to ensure that all meshes at the transition can be connected correctly.

The present study proposes an approach for generating high-quality solid meshes for FEA applications that is based on feature recognition. Although feature recognition has long been studied, its application in FEA has not been investigated extensively. In particular, this study focuses on the development of a rib recognition algorithm, the output of rib data for meshing, and the development of a process for automatic rib meshing. Because ribs are usually formed as a series of structures residing on a part, the proposed algorithm involves a decomposition strategy to separate a rib structure into a set of regions, and the evaluation of the data required for all regions. First, the attributes for ribs and the rules for satisfying ribs are determined. Next, a search is conducted to find all ribs. The data for each rib is essentially a collection of shell, end, and base faces; some of these may be repeated, as they are shared by different ribs. Finally, all rib regions are computed individually. Two basic patterns, namely "rib segment" and "transition region," are defined in this study to separate a rib structure into regions. All rib segment regions are evaluated, followed by all regions belonging to the transition region. The rib decomposition algorithm primarily evaluates a set of slicing faces for each region, which can then be converted into hexahedral or prismatic meshes depending on its type. Each slicing face can serve as the transition between the rib extracted and the part remaining on the CAD model. Several examples are presented to demonstrate the feasibility of the proposed rib recognition and decomposition algorithm.

3. Rib definition

We categorize ribs as either "single ribs" or "rib structures." As Fig. 1(a) depicts, a single rib stands alone on the CAD model. It can be decomposed individually as a region and easily converted into hexahedral meshes. A rib structure, however, is formed by a series of ribs that connect together. Its geometry is complex, and it must be decomposed into separate regions so that each of them can be meshed individually. Rib structures are divided into regions as either "rib segments" or "transition regions" as shown in Fig. 1(a). A rib segment has a pair of matching faces that are symmetric or almost symmetric, and can easily be converted into hexahedral meshes by a mapping method. A "transition region" is the junction of several rib segments. As the connecting conditions of rib segments at a junction can vary significantly, each transition region must be separated from the rib structure and meshed individually. In the proposed algorithm, the rib segments in a rib structure are recognized first, and then the transition regions are detected in accordance with the rib segment data. Fig. 1(b) depicts the basic attributes of a rib segment, including two end faces, shell faces, and base faces. The two end faces represent the matching faces of the rib segment, which are parallel or almost parallel to each other. The number of shell faces and base faces can vary. Also, a fillet may occur at the transition of two adjacent faces; in Fig. 1(b), fillets are shown to have occurred at all adjacent faces.

Fig. 2 classifies various types of single ribs and rib structures. For a single rib, the form of the shell face can vary (Fig. 2[a]). The "single shell face" case is the simplest, and the "multiple shell faces" case is the most complex because "virtual faces" have occurred on the rib. A virtual face is one that covers multiple regions in a CAD model. To decompose a region with virtual faces, its boundary edges and faces at virtual faces must be analyzed and computed. In the case of a "curved shell face," an extension of the rib recognition algorithm from planes to surfaces is required. For these three cases, each rib can be decomposed into a rib segment alone; no transition region is needed. Fig. 2(b) and 2(c) depict rib structures; in Fig. 2(b), the shell faces at the junctions are of the same height, whereas in Fig. 2(c) the shell faces at the junctions vary in height. The intersection of the horizontal and vertical ribs can be an X-connection, T-connection with overlapping, or T-connection without overlapping. It is noted that in the 3rd column of Fig. 2(c), two T-junctions could be very close to each other to result in a tiny gap. The tiny gap should not be separated as an independent region as the meshes generated from this region

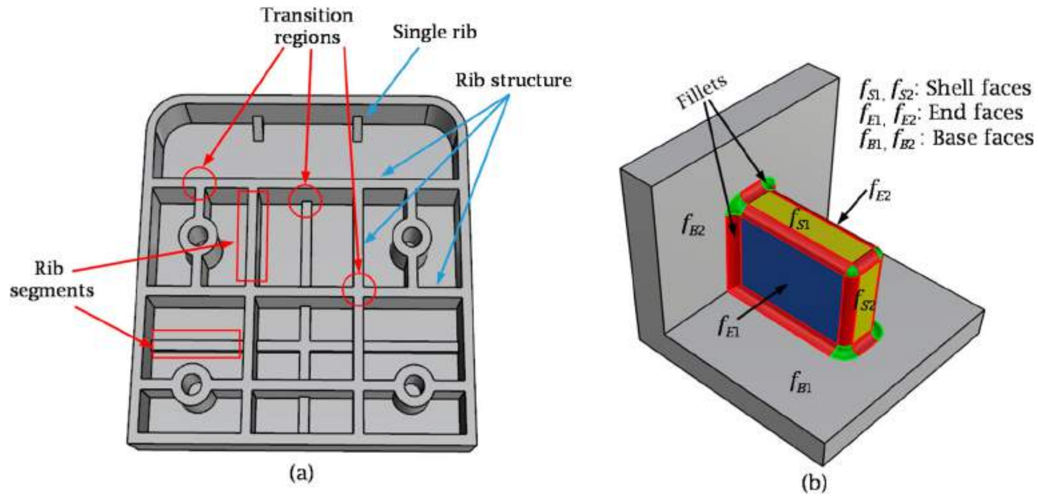

Figure 1. Basic structure of ribs, (a) two basic patterns: rib segment and transition region, and (b) three types of adjacent faces on a rib segment: shell faces, end faces and base face.

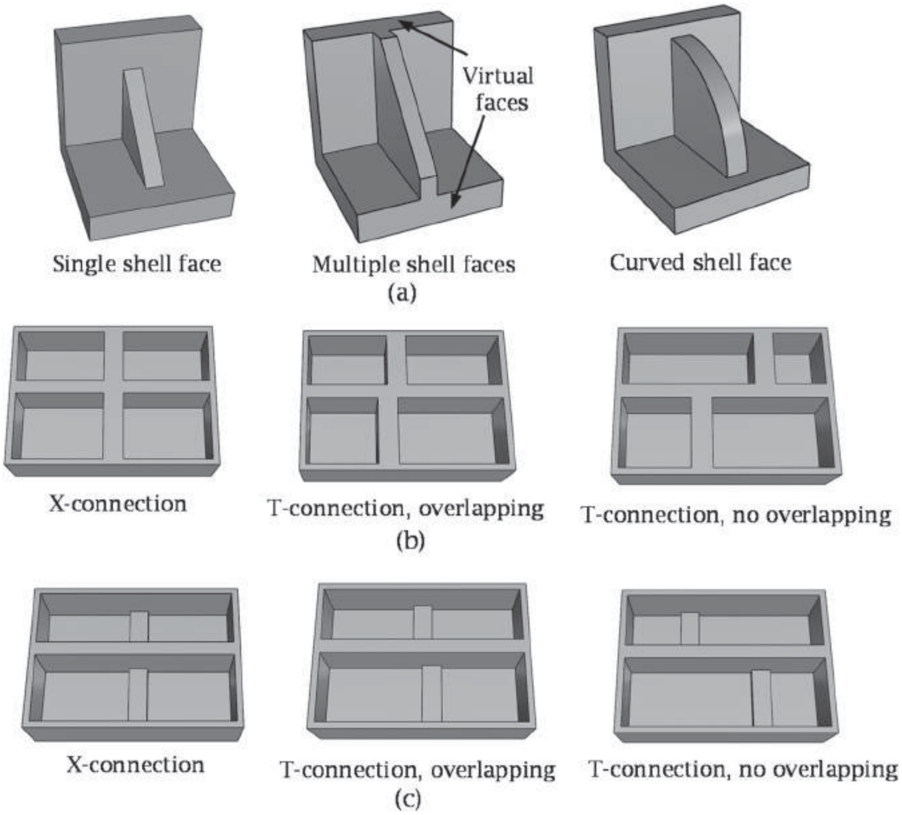
Virtual faces
Single shell face
Multiple shell faces
Curved shell face
(a)
X-connection
T-connection, overlapping
T-connection, no overlapping
(b)
X-connection
T-connection, overlapping
T-connection, no overlapping
(c)

Figure 2. Different situations of ribs, (a) three types of single ribs, (b) three types of junctions with equal height, and (c) three types of junctions with different height.

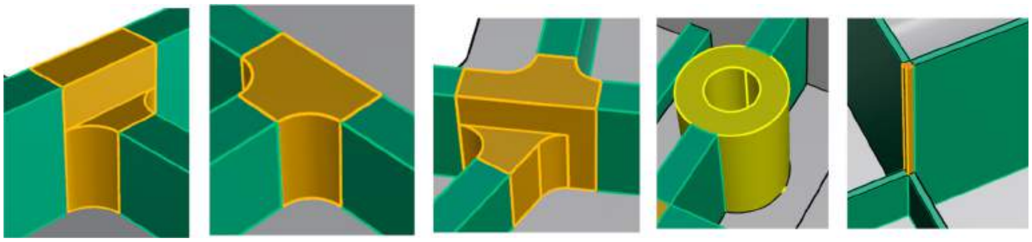

Figure 3. Other connecting conditions at a junction.

could be smaller than what is allowed. Fig. 3 depicts several complex transition regions separated from a variety of rib structures. The meshing method for each of them must be developed separately.

4. Rib recognition and decomposition

Fig. 4 is a flowchart of the proposed feature recognition and decomposition method for ribs, where the input is the B-rep model of an object and the output is the data of all rib regions and the corresponding slicing faces for each of them. The user must assign the maximum thickness allowed (t_{max}) for ribs. The proposed method is divided into three parts: preliminary functions, rib recognition, and rib decomposition. The objective of preliminary functions is to prepare two databases for rib recognition and decomposition: an edge AAG database, which records the topological and geometric information of all edges, and a face AAG database, which records the same information for all faces. The objective of rib recognition is to find all rib segments. A single rib is essentially a rib segment, which is constructed by a pair of end faces that are parallel or almost parallel to each other. A rib structure, however, is composed of rib segments and transition regions. The set of shell faces and base faces adjacent to both end faces are also computed. The rib recognition algorithm is equivalent to a search for the end, shell, and base faces of all rib segments. The objective of rib decomposition is to compute all regions so that they can be decomposed one by one. Each region is either a rib segment or a transition region. The faces that cross distinct rib segments are divided and regenerated so that

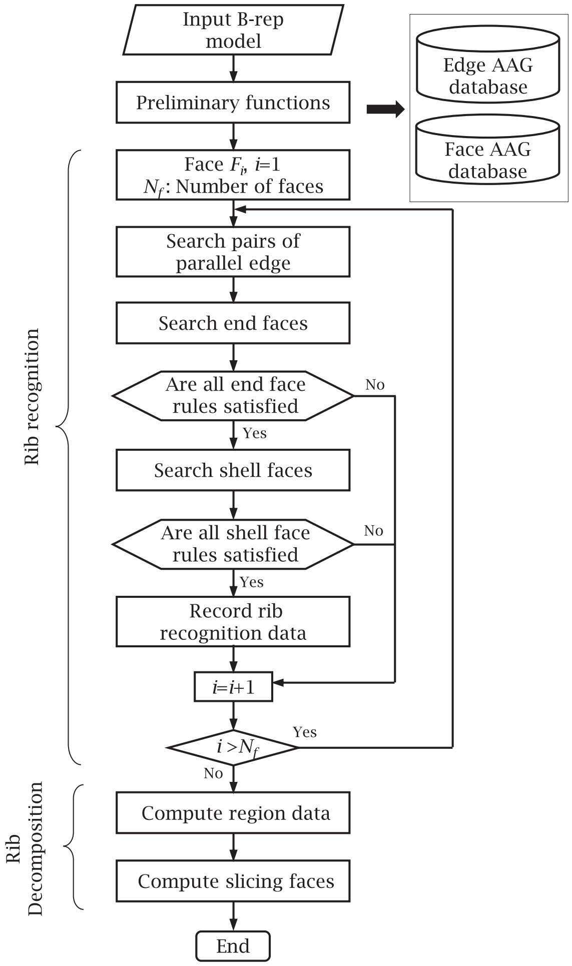

Figure 4. Flowchart of the proposed rib recognition and decomposition algorithm.

each rib segment has its own faces, and the data for all transition regions are also computed. Finally, a set of slicing faces for each region is evaluated so that all regions can be separated from the CAD model.

4.1. Preliminary functions

Preliminary functions establish the data required for the edge and face AAG databases. Three main functions are involved in preliminary functions, i.e. entity grouping, fillet recognition and loop recognition. The purpose of entity grouping is to search edges, planes, and surfaces that should be considered as a group. Entity grouping can be divided into three parts: plane grouping, edge grouping and surface grouping. Plane grouping is employed to group all adjacent planes with the same surface normal. Edge grouping is employed to group all adjacent edges on a face with G^1 continuity. Surface grouping is employed to group all adjacent surfaces with the same curvature. Fillet recognition is employed to recognize all fillets on a CAD model and loop recognition is employed to detect all kinds of loops on the same model. All data obtained are saved in the corresponding attributes of the edge and face AAG databases. If an edge, a plane or a surface is divided into several sub-entities in the CAD data, the grouping data recorded in these two databases can be employed to check all sub-entities of the same group simultaneously. A detailed description for each of the aforementioned functions is given in [28].

The edge AAG is shown in Tab. 1, which contains seven attributes. The first five attributes represent the topological and geometric properties of the edge itself, the sixth attribute records its group index, and the seventh attribute records its loop index. The second attribute denotes the convexity property of the edge, which can further be divided into 11 subtypes (Tab. 1, note 1). For the sixth attribute, all adjacent edges on a face with G^1 continuity should be considered as a group. The topological elements in a B-rep model include the vertex, trim, edge, loop, and face. Trims and edges represent the boundary profile of a face in the parametric and 3D domains, respectively. The distinction between them is that edges are nondirectional whereas trims are directional. Therefore, an algorithm is employed to group trims and edges on the same face when they are G^1 continuous and save the data collectively. For the seventh attribute, a loop is essentially formed by trims. Three types of loops can be defined in a B-rep model: single, virtual, and multivirtual. A single loop is the current loop recorded in the B-rep data structure. A virtual loop lies across faces that are at least G^1 continuous. Finally, a multivirtual loop lies across faces that are either G^0 or G^1 continuous. The proposed loop structure enables the search of loops both within a face and across multiple faces [11].

The face AAG is shown in Tab. 2, which contains eight attributes. The first six attributes represent the topological and geometric properties of the face itself. The seventh attribute records the group index of the face, and the last

Table 1. Edge AAG database.

No.	Attribute	Remark
1	Edge index	Index of the edge
2	Convexity property*	Convexity status of two faces neighboring the edge
3	Convexity angle	Angle of two faces neighboring the edge
4	Geometry type	Line or curve
5	Adjacent faces	Indices of two neighboring faces
6	Group index**	Index of the group to which the edge belongs
7	Loop index***	Index of the loop to which the edge belongs

*11 convexity properties: Non-smooth concave or convex edge, plane-surface concave or convex edge, surface-surface concave or convex edge, sphere concave or convex edge, plane edge, reflection edge, and free edge
**0: Does not belong to any group; Others: Group index
***Can be single, virtual or multi-virtual loop

Table 2. Face AAG database.

No.	Attribute	Remark
1	Face index	Index of the edge
2	Geometry type	Plane or surface
3	No. of loops	Number of loops on the face
4	No. of edges	Number of edges on the face
5	No. of convex edges	Number of convex edges on the face
6	No. of concave edges	Number of concave edges on the face
7	Group index*	Index of the group to which the face belongs
8	Blend-face type*	The type of blend face

*0: Does not belong to any group; Others: Group index
**0: Is not a blend face; 1: EBF; 2: Single VBF; 3: Multiple VBFs

Table 3. Attributes of a rib segment.

Code	Attribute		Remark
R1	Rib index		Index of the rib
R2	Shell face index		Index of the shell face
R3	End face index		Index of the end face
R4	Base face index		Index of the base face
R5	Fillet face index		Index of the fillet face
R6	Information on pair edges	Edge indices	Indices of two edges, each of which is adjacent to a shell face and an end face
R7		Vertices of edges	Two vertices of each edge in R6
R8		Link edges	Two edges connected to the edges in R6
R9		Vertices of edges	Two vertices of each edge in R8
R10	Thickness		Thickness of the rib

attribute records the type of the blend face, if applicable. The third attribute denotes the number of loops on a face, only recording the number of loops on a face in the B-rep model. The seventh attribute denotes the group index of a face. All adjacent surfaces with the same curvature should be considered as a group, which is achieved by checking the curvature of the common edge of two neighboring surfaces. A particular application of this algorithm is when a fillet is divided into two surfaces in the CAD data. When these two surfaces are grouped, they can be considered the same fillet in fillet recognition. The eighth attribute denotes blend-face type. Blending is a common function used in 3D CAD modeling to smooth sharp edges. Typical blend faces are edge blend faces (EBFs) and vertex blend faces (VBFs). If the feature before surface blending is an edge, the blend face is called an EBF, whereas if the feature before surface blending is a vertex, the blend face is called a VBF. A VBF can exist alone or as several that are connected to each other. Other mixed blend faces may also exist [6],[13],[27], but are not considered here as they do not affect rib recognition. A fillet recognition algorithm was also proposed to identify various types of fillets, recording the topological relationships and attributes of edges and faces related to fillets [12].

4.2. Rib recognition

In rib recognition, all faces on the CAD model are tested in sequence to search for rib segments. Tab. 3 lists the attributes of a rib segment that should be recorded. The codes R6–R9 record pair edge information. R6 and R8 record a pair of matching edges, where R6 denotes the edges connected to a shell face and an end face, and R8 denotes the edges connected to the edges in R6. Because the two end points of both parallel edges (R6 and R8) may not be aligned, the corresponding mapping points should also be recorded. R7 records the two end points of R6, and R9 records the corresponding mapping points on R8. The proposed rib recognition algorithm includes three major procedures: searching for pairs of parallel edges, searching for end faces, and searching for shell faces. If all three procedures succeed, a set of rib segments can be recognized and the Tab. 3 attributes for each of the rib segments computed. By contrast, if any of the three procedures fails, no rib segment exists for the face tested. This process is repeated continuously until all faces on the CAD model have been tested. For all three procedures discussed below, F_i represents the face tested.

4.2.1. Search for pairs of parallel edges

The end faces of a rib segment are a pair of matching faces that are parallel or almost parallel to each other. Normally, the two end faces of a rib segment are parallel to each other. However, some end faces may not be completely parallel to each other (e.g., a draft angle designed on the model). Therefore, a small tolerance in the angle between two parallel faces is allowed in the proposed algorithm. The objective of this step is to find multiple pairs of parallel edges for F_i. Each pair of edges must cover one rib segment only. If an edge covers more than one rib segment, it must be divided into several segments, and the corresponding edge segment is recorded.

All parallel edges on F_i are grouped first, using the directional vector of an edge. Consider that the directional vectors of two edges are $\boldsymbol{n}_{ei}$ and $\boldsymbol{n}_{ej}$, respectively. If $|\boldsymbol{n}_{ei} \cdot \boldsymbol{n}_{ej}| \geq k$, where $k = \cos\theta$ and θ is a small angle, then the two edges are considered parallel (or almost parallel) to each other and put into the same group. Here θ is 5°. Once all edges on F_i are tested, several groups of parallel edges can be obtained. For example, the face F_i in Fig. 5(a) is a virtual face that lies across several ribs. There are 17 edges in F_i, divided into two groups: horizontal edges and vertical edges. Note that some of the horizontal edges are not well aligned. The longer edges should be subdivided so that each of the shorter edges can find its matching segment with both end points aligned.

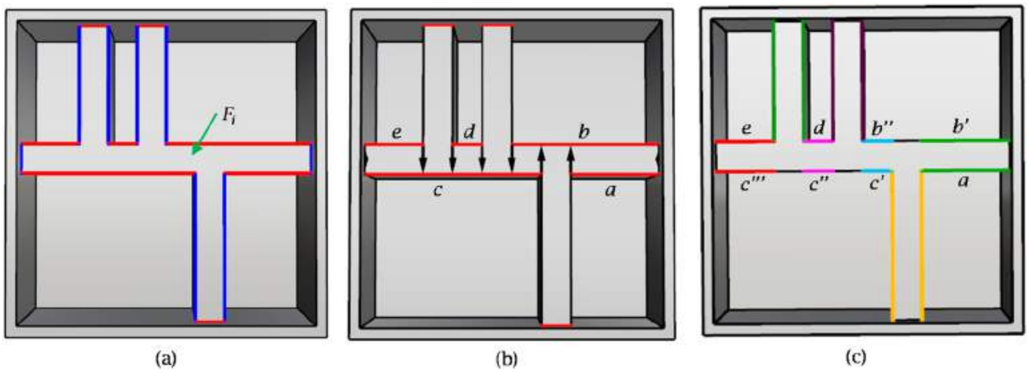

Figure 5. Search pairs of parallel edges, (a) grouping of parallel edges, (b) evaluation of horizontal pairs of edge segments, and (c) final pairs of edge segments, where seven pairs of parallel edges are found.

On each group of parallel edges, an algorithm is implemented to divide them into pairs. For each end point of the edges, the matching edge with the minimum distance to this point is computed. Once all matching edges are obtained, effective pairs of matching edges can be evaluated. When projecting an end point onto its edge pair, the system will check the distance between the projected point and its closest end point on the edge pair. If the distance is smaller than an allowable tolerance, say the smallest mesh size allowed, this end point is considered as the projected point. It can avoid the occurrence of tiny regions for the case in the 3rd column of Fig. 2(c). If the two end points of an edge map onto the same edge, this pair of edges is an effective pair. By contrast, if the two end points map onto two different edges, it is an ineffective pair. For example, for the group of horizontal edges in Fig. 5(b), $a^s \to b$, $a^e \to b$, $b^e \to c$, $c^s \to b$, $d^s \to c$, $d^e \to c$, $e^s \to c$, and $e^e \to c$, where the superscripts s and e denote the starting and end points of an edge, respectively, and the symbol → denotes the mapping (for example, $a^s \to b$ means that the matching edge of the starting point of a is b) For each mapped pair, if only two edges appear, then it is effectively a pair of parallel edges. Therefore, four effective pairs of edges are found in this example, namely $a \to b$’, c’ $\to b$”, c” $\to d$, and c’” $\to e$, where all intersection points are already computed. Fig. 5(c) depicts the result of searching for pairs of parallel edges in the example of Fig. 5(a).

4.2.2. Search for end faces

A rule-based search algorithm is proposed to determine the end faces for each pair of parallel edges associated with F_i. Two faces neighboring the two parallel edges are obtained by using the edge AAG database. If either of them is an EBF, the face AAG database is employed to pass over it and obtain the correct candidate face. Once a pair of candidate faces is obtained, the following six rules are employed to check if they satisfy the conditions of end faces. F_c represents a candidate face under consideration.

(1) *Rule 1- Both candidate faces should be planes*: Most end faces on ribs are planes. A standard meshing algorithm can be employed to convert such a rib segment into hexahedral meshes. All of the subsequent rules are based on this prerequisite. If both side faces of a rib segment are curved surfaces, the rules below can still be applied if all attributes are evaluated in accordance with curved surfaces.

(2) *Rule 2- The number of concave edges on F_c must be larger than 1*: F_c must have more than one concave edge to be an end face. If F_c has only one concave edge, then F_c may be part of an extrusion or a shell face and should not be considered an end face.

(3) *Rule 3- If inner loops exist on F_c, all edges on the inner loops should be convex edges:* As the function of ribs is to enhance the strength of a structure, most end faces on ribs are as simple as possible. If a hole exists on F_c, then the rib segment can still be converted into hexahedral meshes. By contrast, if an extrusion exists on F_c, then the shape is more complex than it would have been otherwise and is therefore not considered a rib segment.

(4) *Rule 4- Two base faces neighboring two distinct concave edges should be adjacent to each other*: A rib should connect at least two base faces to enhance the strength of a structure. Even if two concave edges on F_c are not connected, F_c is still considered a candidate end face if two base faces are connected. By contrast, if two base faces corresponding to two distinct concave edges are not adjacent to each other, then F_c is not an end face.

(5) *Rule 5- Two candidate faces F_c are not adjacent to each other*: If two candidate faces F_c are adjacent to each other, the grouping of parallel edges might be wrong owing to some unexpected conditions of the CAD mode.

(6) *Rule 6- The effective angle of two candidate faces F_c is 180 ± ε°*: If two candidate faces F_c are parallel to each other, then two normal vectors should ideally be 180°. However, the angle can be changed slightly by considering actual rib design in CAD models. Here, ε could be adjusted by the user, and the default is set to 8°.

These tests should be performed for both candidate faces F_c. If all six rules are satisfied for both candidate faces, then these are regarded as the end faces of F_i. Otherwise, no end faces are found for F_i. The attributes R3, R5, and R6 in Tab. 3 can directly be recorded when a pair of end faces is obtained, and the attributes R7 to R9 can be computed using the F_i data.

4.2.3. Search for shell faces

The faces that are adjacent to a pair of end faces include shell and base faces (Fig. 1[b]). F_i in Section 4.2.2 is one of the shell faces. As many shell faces may exist on a rib segment, as shown in Fig. 1(b), the end faces must first be found from F_i, and then all shell faces must be obtained from the end faces. The edge between an end face and a base face is concave, whereas the edge between an end face and a shell face is convex. This property is employed to distinguish the type of face that neighbors an end face. When a fillet exists between any of these two end faces, the face AAG database can be employed to pass over the fillet and obtain the correct neighboring face.

The procedure to search for shell faces from a pair of end faces (f_{E1} and f_{E2} in Fig. 1(b)) is performed by checking all edges for one of the two end faces, say f_{E1}. Consider that the edge to be tested on f_{E1} is e_j. First, the neighboring face of e_j is found and whether e_j is a convex edge is checked. If it is, the process continues; if not, e_j is discarded and the next edge on f_{E1} is found. Second, whether the neighboring face of e_j is a fillet is determined. If it is, the face AAG database is employed to obtain the correct neighboring face; if not, the neighboring face can be obtained directly. Third, whether the neighboring face connects to the other end face (i.e., f_{E2}) is determined. If it does, this face is recorded as one shell face; if not, e_j is discarded and the next edge for the test is found. This process is continued until all edges on f_{E1} have been tested. Once this process is complete, all shell faces related to the pair of end faces f_{E1}–f_{E2} can be obtained. Two attributes in Tab. 3, namely, shell face index (R2) and base face index (R4), can be recorded in this stage.

4.3. Rib decomposition

The data recorded in Tab. 3 represent the edges and faces involved in each rib segment, but they are still insufficient to describe an individual region because some of the entities exist across multiple regions. Furthermore, the data to describe transition regions have still not been considered. Therefore, the purpose of rib decomposition is to compute the data required for decomposing each region, both rib segments and transition regions. The region data are primarily composed of two parts: boundary edges and slicing faces. The former describes the transition edges of a region and the rest of the model, and the latter describes the faces required to decompose a region.

The proposed rib decomposition algorithm should handle two types of rib structures: those of equal height (Fig. 2[b]) and different height (Fig. 2[c]). Rib structures combining both may also exist. In the proposed method, the shell face on a rib structure is divided into loops that represent the regions that should be decomposed. As there are two types of height, equal height (Fig. 2[b]) is analyzed to yield a set of loops, and then different height (Fig. 2[c]) is analyzed for each. If different height exists in a loop, the loop is further subdivided. Once all loops are obtained, the boundary edges and slicing faces for each region can then be computed. Fig. 6 is a flowchart of the proposed rib decomposition algorithm, which can be divided into the following four steps: (1) divide all shell faces into loops, (2) subdivide the loops and compute boundary edges for all rib segments, (3) compute boundary edges for all transition regions, and (4) compute slicing faces for all regions. Each step is described below.

4.3.1. Step 1: Divide all shell faces into loops

A loop is formed by four edges that connect to each other in sequence. The loops on a shell face come from both types of region, rib segments and transition regions. The pairs of edges on a shell face are employed to evaluate the loops, with respect to rib segments first. The edges that link each pair of edges are also generated. The loops arising from transition regions are then computed using the link edges. Fig. 7 depicts the procedure in detail. First, the loops for rib segments are created. For each pair of edges, every two end points are connected to form a link edge. The four edges are arranged in sequence to create a loop. When all pairs of edges are processed, the loops for all rib segments are created. All link edges are then grouped. Fig. 7(a) indicates the generation of a loop and Fig. 7(b) indicates the loops for all rib segments sharing the same shell face. Second, the edges for the transition regions are created. Each edge on the shell face is compared with the pair edges. The segments that do not belong to any pair edge are detected and new edges are created. These edges are also included in the group of link edges. Fig. 7(b) depicts a new edge created for a transition region. Third, the loops for the transition regions are

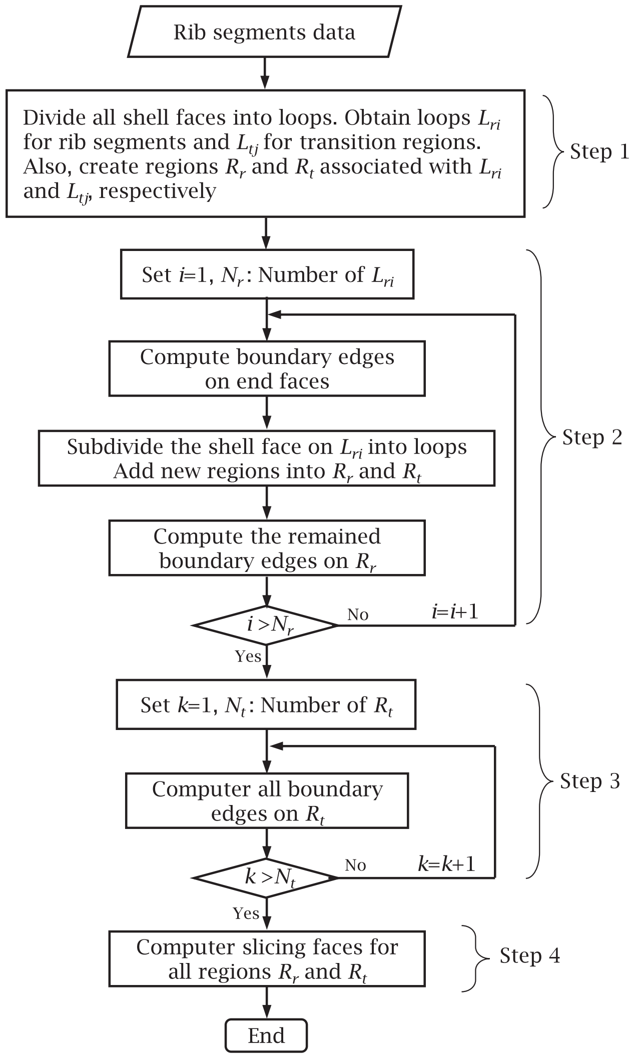
Rib segments data
Divide all shell faces into loops. Obtain loops L_{ri} for rib segments and L_{tj} for transition regions. Also, create regions R_r and R_t associated with L_{ri} and L_{tj}, respectively
Step 1
Set i=1, N_r: Number of L_{ri}
Compute boundary edges on end faces
Subdivide the shell face on L_{ri} into loops Add new regions into R_r and R_t
Compute the remained boundary edges on R_r
$i > N_r$
No
i=i+1
Yes
Step 2
Set k=1, N_t: Number of R_t
Computer all boundary edges on R_t
$k > N_t$
No
k=k+1
Yes
Step 3
Computer slicing faces for all regions R_r and R_t
Step 4
End

Figure 6. Flowchart of the proposed rib decomposition algorithm.

created. All edges on the group of link edges are tested. Whenever a series of edges can be connected in sequence to form a loop, a loop for a transition region is created (Fig. 7[c]). The output of this step is comprised of loops $L_{ri,i=1\ldots Nr}$ for rib segments and loops $L_{tj,j=1\ldots Nt}$, for transition regions. Regions R_r and R_t, associated with L_{ri} and L_{tj}, respectively, are also created (Fig. 6). The number of regions in R_r and R_t increase as the process continues.

4.3.2. Step 2: Subdivide loops and compute boundary edges for all rib segments

A loop must further be subdivided at the junction of different heights so that the shape at the junction can be recognized and decomposed as an individual region. Although these rib segments do not share the same shell face, they still lie on the same base face. Therefore, the pair edges of these rib segments on the base face are employed for detecting and subdividing such a loop. The proposed subdivision procedures are as follows. First, for a rib segment, the edges that are adjacent to both an end face and the base face are found. If one edge also lies on a shell face, the segment that lies on the shell face should be detected and removed. In Fig. 8(a), the loop L_{ri} on the shell face denotes the rib segment considered, and the bold lines are edges adjacent to both an end face and the base face. Second, each edge on one end face is checked to find its matching edge on the other end face. Each pair of matching edges is counted once. In Fig. 8(b), three pairs of matching edges were detected. Third, each end point of an edge on an end face is projected onto the other end face and whether it can connect to an existing vertical edge is assessed. All vertical edges found are included in the edge pairs, as shown in Fig. 8(c). Fourth, each of the new vertical edges found on one end face is projected onto the other end face to generate a new pair of edges. Each of the new edges is extended so that it intersects the shell face, and the edges on the loop are subdivided. Fig. 8(d) depicts the four pairs of vertical edges found. Finally, the loop is subdivided in terms of its new subdivided segments. A link edge for each pair of edges is generated first, and then every four edges are connected in sequence to form a new loop. In the case of Fig. 8(e), three loops are generated. This process is repeated for all rib segments. In Section 4.3.1, a loop was generated for each rib segment. Whenever a loop is subdivided in this step, it is removed

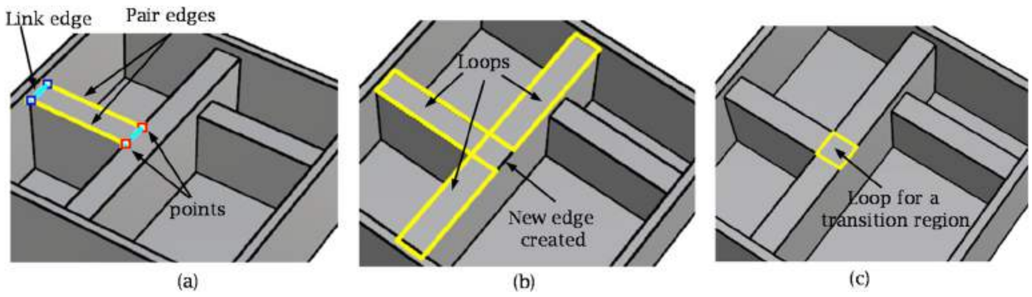

Figure 7. Divide a shell face into loops, (a) a loop created from a pair of edges, (b) all loops created from all pairs of edges, and (c) a loop created for the transition region.

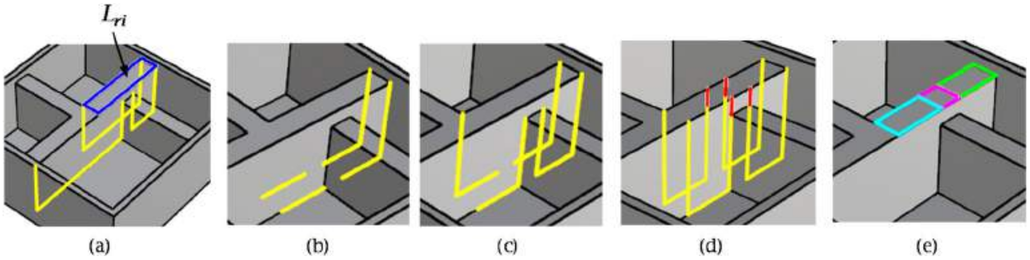

Figure 8. Subdivide a loop, (a) all edges adjacent to one end face and the base face simultaneously, (b) pairs of matching edges are computed, (c) Project each end point to find an existing vertical edge on the other end face, (d) find pairs of vertical edges and intersect the shell face, and (e) obtain the subdivided loops.

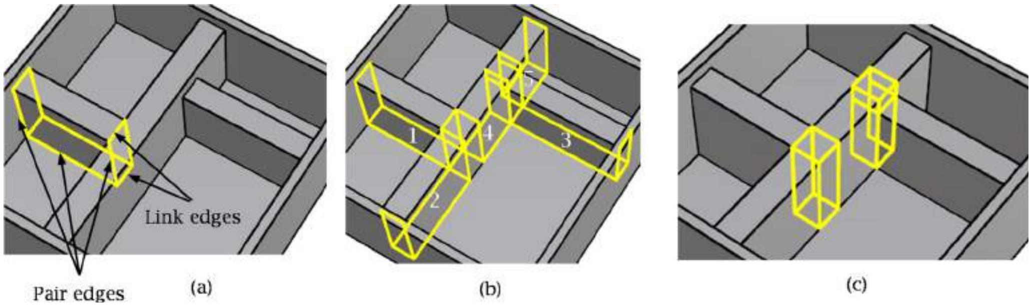

Figure 9. Compute boundary edges, (a) boundary edges for a rib segment, (b) boundary edges for all rib segments sharing the same shell face, and (c) boundary edges for all transition regions sharing the same shell face.

and the subdivided loops are added. When a loop is created, a new region is created also. The regions R_r for rib segments and R_t for transition regions are the output of this step (Fig. 6).

The data on each region include boundary edges and slicing faces. The boundary edges are used to generate the slicing faces and the slicing faces are used to decompose a region. As Fig. 9(a) depicts, each slicing face can be formed by a series of boundary edges that from a loop. Most of the boundary edges corresponding to a rib segment are already obtained, such as in Fig. 8(d), where the edge pairs at the transition of the rib segment and the base face are actually part of the boundary edges. It is only necessary to generate link edges to connect each pair of edges and form the required loops for slicing faces. Fig. 9(b) depicts five sets of boundary edges for five segments, respectively.

4.3.3. *Step 3: Compute boundary edges for all transition regions*

The boundary edges corresponding to the regions R_t are computed. Each loop on a transition region contains at least four edges. If an edge is adjacent to a rib segment, the corresponding boundary edges can be obtained by using those from the neighboring rib segment. By contrast, if an edge is not adjacent to a rib segment, each of its two end points is still adjacent to a rib segment. Therefore, each of the end points can be used to find the corresponding boundary edges by using those from the neighboring rib segment. All boundary edges can then be arranged in sequence to form the loops required for slicing faces. Fig. 9(c) depicts two sets of boundary edges obtained for two transition regions.

4.3.4. *Step 4: Compute slicing faces for all regions*

Once all boundary edges are obtained, every four boundary edges are arranged in sequence to form a loop. Given that all slicing faces are planes, each loop of four boundary edges can be used to generate a planar face. This process is repeated for all loops to yield all slicing faces. The slicing faces corresponding to each region can be grouped and employed later for the decomposition of the region.

5. Results and discussion

We tested feasibility with a program based on the proposed algorithms, written in C++ and based on the Rhino CAD platform [25] and the openNURBS function [23]. The proposed algorithm only deals with manifold CAD model. Also, if the CAD model is topologically or geometrically incorrect, it should be corrected first. As Rhino Fig. 10 illustrates examples of the rib recognition results obtained; the ribs are shown in blue. Shape complexity ranged from simple individual ribs to several interconnected rib structures. Ribs recognized

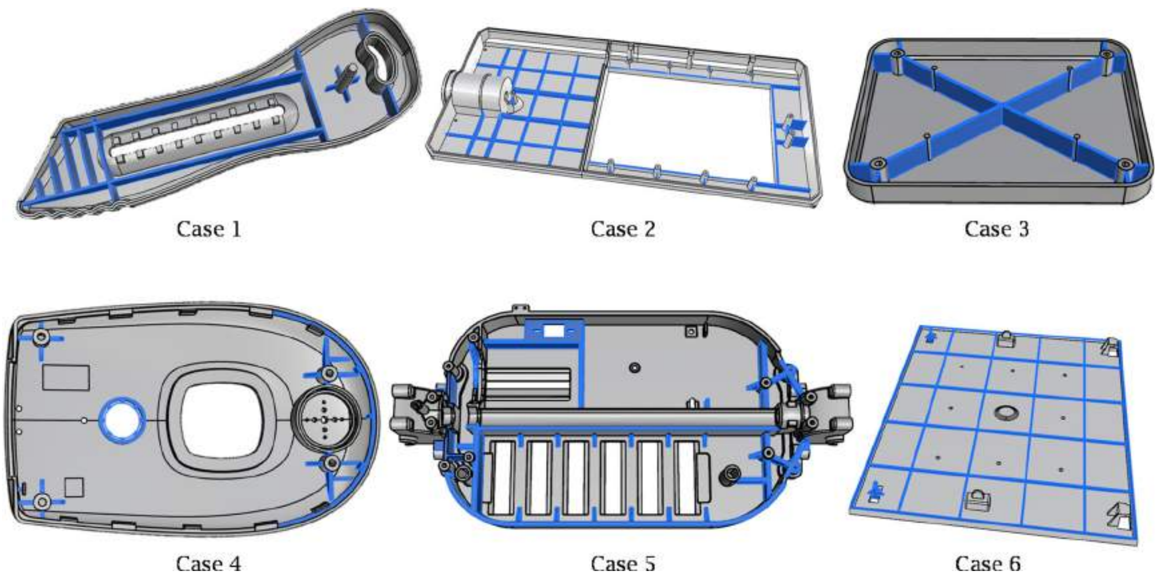

Figure 10. Results of the proposed rib recognition algorithm for six CAD models.

successfully in this study include rib structures located on several surfaces or irregular faces, and a rib structure with a mixture of segments of equal and different heights.

In addition, we compared all results with those obtained using CADdoctor^{TM} [4] under the same conditions for the cases in Fig. 10. Tab. 4 compares the parameters used and the recognition results for our study with those of CADdoctor, where t_{max} denotes the maximum rib thickness allowed (the default value was used for all cases). As the results indicate, the rib definitions in our study are slightly different from those of CADdoctor. For example, a rib structure can be decomposed into many rib segments in the proposed algorithm, whereas it is not in CADdoctor. This indicates that our rib recognition is more flexible than CADdoctor, although it still has limitations. For instance, the following cases have not been considered in the proposed algorithm: (1) when a rib has no base face, and (2) when the end faces are curved surfaces. These two cases need to be studied further as they may be mixed with other extruded types of features. The misjudgment regions for cases 4 and 5 by using CADdoctor are shown in Figs. 11(a) and 11(b), respectively. The regions pointed by an arrow in 11(a) and 11(b) are wrongly recognized as ribs in CADdoctor.

Table 4. Results of rib recognition for six CAD models, where success and fail denote the number of rib segments recognized successfully and unsuccessfully, respectively, and misjudgment denotes the number of features that are wrongly regarded as ribs.

			Recognition results Success/ Fail/Misjudgment	
Case	No. Ribs/ Total faces	*tmax* (mm)	This study	CADdoctor^{TM}
1	18/369	10	18/0/0	18/0/2
2	48/726	10	48/0/0	48/0/6
3	16/161	10	16/0/0	16/0/0
4	32/411	10	32/0/0	32/0/1
5	38/3385	10	38/0/0	35/3/6
6	46/324	10	46/0/0	46/0/2

Fig. 12 illustrates the immediate results of rib decomposition for the first CAD model. As the original CAD model (Fig. 12[a]) shows, the rib structure contains a junction with equal height. By assessing the two shell faces individually, five loops are obtained, including four rib segments and one transition region. Subsequently assessing the base face for ribs of different height divides one loop into three sub-loops (Fig. 12 [b]). The final total of seven loops is used to compute the boundary edges for all seven regions. Every four connecting edges form a slicing face, and the final seven sets of slicing faces are depicted in Fig. 12(c). Each set can be used to separate a rib region from the CAD model.

However, this example shows a simple case where the two T-connections do not overlap; the two transition regions obtained are simple rectangular solids, which can be meshed using hexahedral meshes as shown in (Fig. 13[a]). When two T-connections overlap (e.g., as in Fig. 13[b] and 12[c]), the shape of the transition regions can be more irregular. In Fig. 13(b), prismatic meshes are used for the transition region because one rib segment is the same height as the transition region; in Fig. 13(c), boundary layer meshes (BLMs) [22] are used for the transition region because the heights of both rib segments vary from that of the transition region. Fig. 13(d) depicts a more complex situation, in which the rib segments do not intersect perpendicularly. Fillets are kept in this case as their diameters are larger than the threshold. The most important issue here is to determine the boundary of the

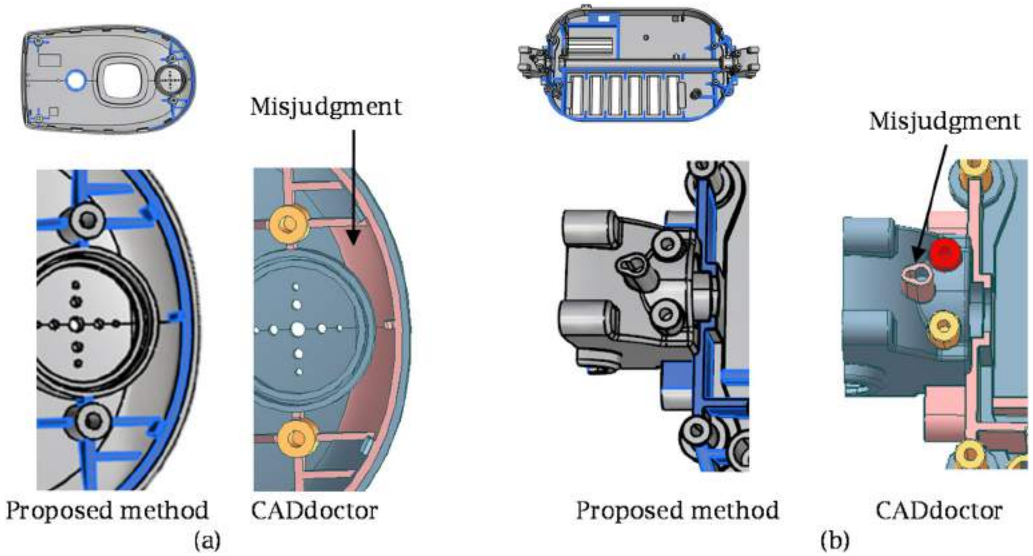

Figure 11. Misjudgment by CADdoctor, (a) Case 4, and (b) Case 5.

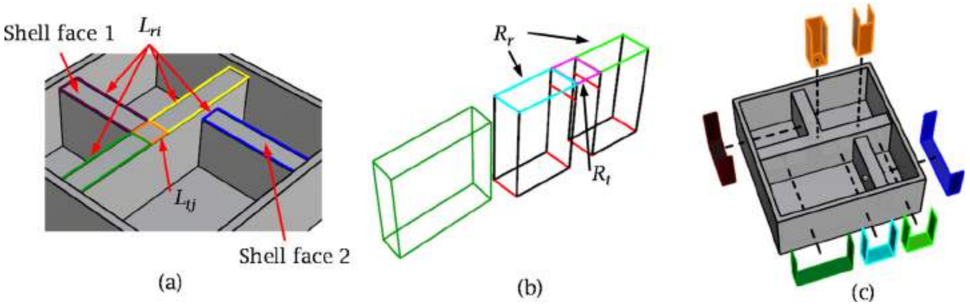

Figure 12. Immediate results of rib decomposition for the first CAD model, (a) the CAD model and five loops obtained by dividing both shell faces, (b) subdivide a loop with two intersected ribs of different height to yield three loops, and (c) seven set of slicing faces generated, corresponding to seven regions, respectively.

transition region and each rib segment. Each boundary is determined along the margin of a fillet so that all shapes related to fillets are assigned to the transition region. Therefore, both rib segment and transition region can be meshed successfully. In Fig. 13(d), prismatic meshes are used for the transition region because both rib segments and transition region are of the same height. All rib segments and transition regions in Fig. 13 are already converted into appropriate types of meshes; it provides a clear description of the mesh structure used for different kinds of junctions.

Fig. 14 depicts the decomposition results for the second CAD model, in which both T- and X- connections exist, as well as rib segments of both equal and different

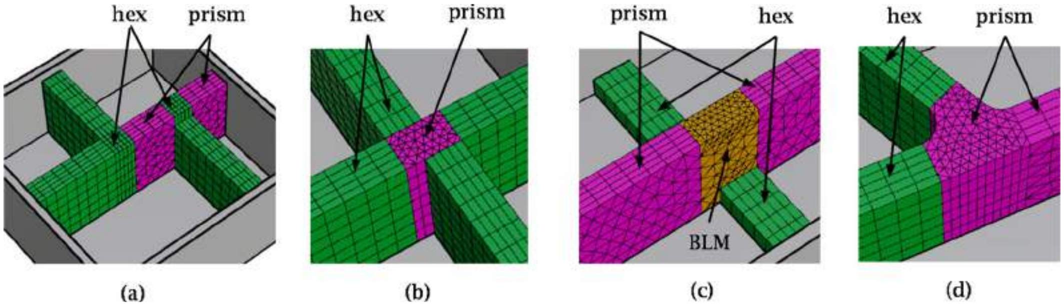

Figure 13. The meshing methods for different kinds of connections at junctions, (a) two T junctions not overlapped, (b) two T junctions overlapped, with different heights on one side, (c) two T junctions overlapped, with different heights on both sides, and (d) a non-perpendicular junction with fillets between rib segments.

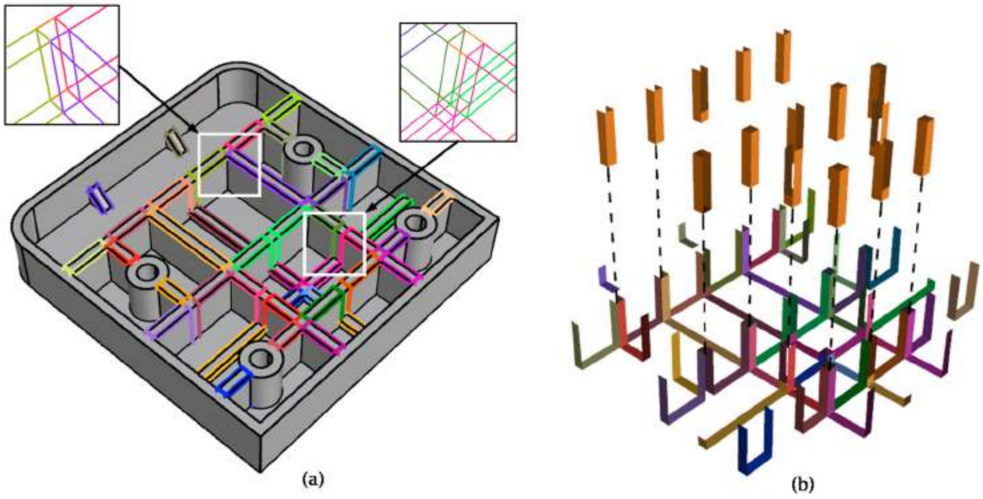

Figure 14. Results of rib decomposition for the second CAD model, (a) the CAD model and boundary edges obtained for all rib segments and transition regions, and (b) all sets of slicing faces for all regions.

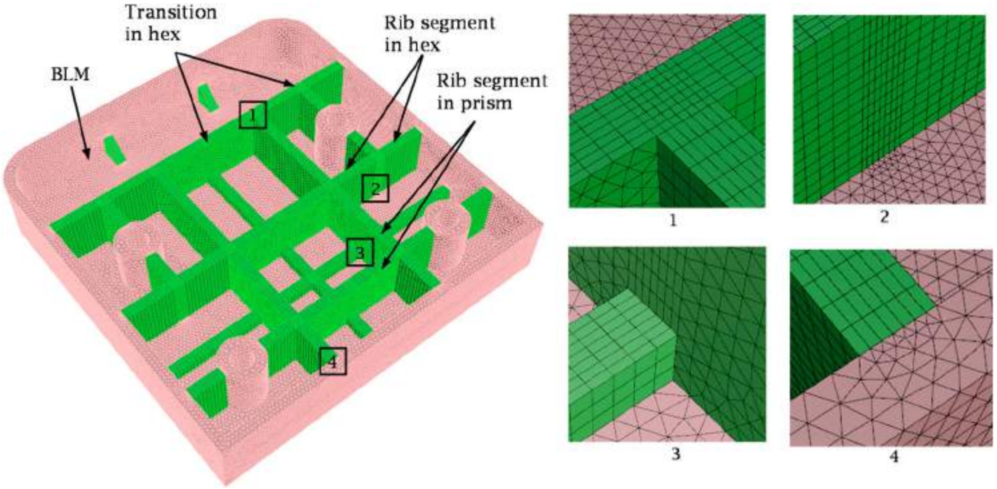

Figure 15. Results of manual mesh generation in accordance with all regions generated in this study, where regular meshes are used for rib regions and BLM meshes are used for the other parts of the model.

height. Four tubes also appear on the model and intersect with the rib segments. The main purpose of this example is to demonstrate the generation of high-quality solid meshes for ribs. Fig. 14(a) depicts the CAD model and the boundary edges obtained, with further details illustrated for two of the junctions. All of the slicing faces are depicted in Fig. 14(b). Note that the slicing faces for the rib segments and transition regions are highlighted differently, and that all boundary edges and slicing faces can be generated automatically. Fig. 15 depicts the solid meshes generated for the case in Fig. 14 by considering all of the rib regions obtained in this study. All of the rib segments and transition regions are meshed using hexahedral meshes, whereas other parts of the model are meshed using BLMs [22]. All quality indices, such as the mesh number, orthogonality, aspect ratio, and skewness, have improved considerably for the ribs owing to the use of hexahedral meshes throughout. However, the indices are not improved for the other parts of the CAD model because the current algorithms are effective for ribs only. It would be necessary to extend the proposed method to handle a greater diversity of features to improve the mesh quality of the entire model.

6. Conclusion

This study focused on the development of rib recognition and decomposition algorithms for thin-walled plastic parts, and verified the feasibility of the proposed algorithms using several CAD models. A procedure for generating hexahedral and prismatic meshes for ribs was also presented. Our approach is based on feature recognition and decomposition for generating high-quality solid meshes for FEA applications. An algorithm for the recognition of a rib structure was proposed that can not only be used for ribs but also expanded to other features. This method can reduce the necessity of manual operation, hence decreasing the overall operational time for meshing. Furthermore, the meshing results indicate that the quality indices of the meshes generated using the proposed method improved considerably in comparison with the results of previous methods. The proposed meshing algorithm is feasible only for recognized ribs, however. Future studies should develop additional feature recognition and decomposition algorithms for converting more CAD model features into hexahedral meshes.

ORCID

Jiing-Yih Lai http://orcid.org/0000-0002-0495-0826
Ming-Hsuan Wang http://orcid.org/0000-0003-4947-4218
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] Ansaldi, S.; Floriani, L. D.; Falcidieno, B.: Geometric modeling of solid objects by using a face adjacency graph representation, ACM SIGGRAPH Computer Graphics, 19(3), 1985, 131–139. http://doi.org/10.1145/325165.325218

[2] 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://doi.org/10.1007/s00366-014-0382-x

[3] 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://doi.org/10.1007/s00366-014-0382-x

[4] CADdoctor, http://elysiuminc.com/products/caddoctor/

[5] 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://doi.org/10.1016/j.cad.2004.02.005

[6] 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://doi.org/10.1080/16864360.2004.10738284

[7] Ismail, N.; Abu Bakar, N.; Juri, A. H.: Recognition of cylindrical-based features using edge boundary technique for integrated manufacturing, Robotics and Computer-Integrated Manufacturing, 20(5), 2004, 417–422. http://doi.org/10.1016/j.rcim.2004.03.004

[8] Ismail, N.; Abu Bakar, N.; Juri, A. H.: Recognition of cylindrical and conical features using edge boundary classification, 2005 International Journal of Machine Tools and Manufacture, 45(6), 2005, 649–655. http://doi.org/10.1016/j.ijmachtools.2004.10.008

[9] Joshi, S.; Chang, T. C.: Graph-based heuristics for recognition of machined features from a 3D solid model, Computer-Aided Design, 20(2), 1988, 58–66. http://doi.org/10.1016/0010-4485(88)90050-4

[10] 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://doi.org/10.1016/j.cad.2014.07.005

[11] Lai, J.-Y.; Wang, M.-H.; Chiu, Y.-K.; Hsu, C.-H.; Tsai, Y.-C.; Huang, C.-Y.: Recognition of depression and protrusion features on B-rep models based on virtual loops, Computer-Aided Design and Applications, 13(1), 2015, 95–107. http://doi.org/10.1080/16864360.2015.1059200

[12] Lai, J.-Y.; You, Z.-W.; Chiu, Y.-K.; Wang, M.-H.; Hsu, C.-H.; Tsai, Y.-C.; Huang, C.-Y.: On the development of blend faces and holes recognition algorithm for CAE applications, Key Engineering Materials, 656–657, 2015, 789–794. http://doi.org/10.4028/www.scientific.net/KEM.656-657.789

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

[14] 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 (ICECE), 2011, 4386–4389. http://doi.org/10.1109/ICECENG.2011.6057432

[15] Li, Y.; Wang, W.; Liu, X.; Ma, Y.: Definition and recognition of rib features in aircraft structural part, International Journal of Computer Integrated Manufacturing, 27(1), 2014, 1–19. http://doi.org/10.1080/0951192X.2013.799784

[16] 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://doi.org/10.1243/09544054JEM1634

[17] Lim, T.; Medellin, H.; Torres-Sanchez, C.; Corney, J. R.; Ritchie, J. M.; Davies, J. B. C.: Edge-based identification of DP-features on free-form solids, 2005 IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 2005, 851–860. http://doi.org/10.1109/TPAMI.2005.118

[18] Lu, Y.; Gadh, R.; Tautges, T. J.: Feature based hex meshing methodology: feature recognition and volume decomposition, Computer-Aided Design, 33(3), 2001, 221–232. http://doi.org/10.1016/S0010-4485(00)00122-6

[19] Lu, Y.; Li, Y.-G.: A feature recognition technology of complex structural parts based on re-extended attributed adjacency graph, Machinery Design & Manufacture, 219(5), 2009, 240–242.

[20] 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://doi.org/10.1007/s00366-012-0302-x

[21] Marefat, M.; Kashyap, R.: Geometric reasoning for recognition of three-dimensional object features, Pattern Analysis and Machine Intelligence, IEEE, 12(10), 1990, 949–965. http://doi.org/10.1109/34.58868

[22] Moldex3D, http://www.moldex3d.com/en/

[23] openNURBS, http://www.rhino3d.com/tw/opennurbs

[24] Owodunni, O.; Mladenov, D.; Hinduja, S.: Extendible classification of design and manufacturing features, CIRP Annals- Manufacturing Technology, 51(1), 2002, 103–106. http://doi.org/doi:10.1016/S0007-8506(07)61476-0

[25] Rhinoceros, http://www.rhino3d.com

[26] Shah, J. J.; Anderson, D.; Kim, Y. S.; Joshi, S.: A discourse on geometric feature recognition from CAD models, Journal of Computing and Information Science in Engineering, 1(1), 2001, 41–51. http://doi.org/10.1115/1.1345522

[27] Venkataraman, S.; Sohoni, M.: Blend recognition algorithm and applications, Proceedings of the sixth ACM Symposium on Solid Modeling and Applications, 2001, 99–108.

[28] Wang, M.-S.; Lai, J.-Y.; Hsu, C.-H.; Tsai, Y.-C.; Huang, C.-Y.: Boss Recognition Algorithm and Application to Finite Element Analysis, Computer-Aided Design and Applications, 14(4), 2017, 450–463. http://doi.org/10.1080/16864360.2016.1257187

[29] 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. http://doi.org/10.1631/%jzus.A0820705

[30] Zhu, H.; Shao, T.; Liu, Y.; Zhao, J.: Automatic hierarchical mid-surface abstraction of thin-walled model based on rib decomposition, Advances in Engineering Software, 97, 2016, 60–71. http://doi.org/10.1016/j.advengsoft.2016.01.007