MBD Based 3D CAD Model Automatic Feature Recognition and Similarity Evaluation

SHUHUI DING\textsuperscript{1,2}, QIANG FENG\textsuperscript{1}, ZHAOYANG SUN\textsuperscript{3}, AND FAI MA\textsuperscript{2}

\textsuperscript{1}College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266590, China
\textsuperscript{2}Department of Mechanical Engineering, University of California, Berkeley, CA 94709, USA
\textsuperscript{3}China National Institute of Standardization, Beijing 100191, China

Corresponding author: Zhaoyang Sun (sunzhy@cnis.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52005302, and in part by the China National Coal Association Foundation under Grant MTK2016-282.

\textbf{ABSTRACT} Automatic Feature Recognition (AFR) is considered as the key connection technique of the integration of Computer Aided Design (CAD) and Computer Aided Process Planning (CAPP). At present, there is a lack of a systematic method to identify and evaluate the local features of 3D CAD models. The process information such as topological structure, shape and size, tolerance and surface roughness should be considered. Therefore, a novel Model Based Definition (MBD) based on 3D CAD model AFR and similarity evaluation are proposed in this paper. A Multi-Dimensional Attributed Adjacency Matrix (MDAAM) based on MBD is established based on the fully consideration of the topological structure, shape and size, surface roughness, tolerance and other process information of the B-rep model. Based on the MDAAM, a two-stage model local feature similarity evaluation method is proposed, which combines the methods of optimal matching and adjacency judgment. First, the faces of source feature and target model are used as independent sets to construct a bipartite graph. Secondly, supplement the vertices in the independent set of source feature to make the number of vertices in two independent sets equal. Thirdly, based on MDAAM data, the weighted complete bipartite graph is constructed with the face similarity between two independent sets as the weight. Fourthly, Kuhn-Munkres algorithm is used to calculate the optimal matching between the faces of source feature and target model. Fifthly, the adjacency between matching faces in target model is judged. Finally, the similarity between matching faces of the two models is calculated, which is used as the similarity evaluation result. The effectiveness of this method is verified by three applications.

\textbf{INDEX TERMS} Automatic feature recognition, similarity evaluation, multi-dimensional attributed adjacency matrix, weighted complete bipartite graph, Kuhn-Munkres algorithm.

\section{I. INTRODUCTION}

Automatic Feature Recognition (AFR) refers to the extraction of feature information with specific engineering semantics from part models. Computer Aided Design (CAD) model of part is composed of low-level graphic elements such as faces, lines, points and etc., while Computer Aided Process Planning (CAPP) information is generated based on high-level features with engineering semantics, such as holes. There is a lack of an intelligent interface for feature recognition between CAD and CAPP. AFR can identify the corresponding structural features according to specific process requirements, which is the basis of process design and the most important step to transform design information into manufacturing process information. With the help of AFR, structure features can be identified according to specific process information, which is considered the most crucial technique to turn design information into manufacturing process [1].

In recent years, research in feature recognition has received significant attention, but the traditional research on feature recognition mainly focuses on the topological structure and face geometry of the model, without considering its process information. If the surface roughness and tolerance level of two models with identical topological structure and geometric dimension have a significance difference, their machining process may be completely different. Therefore, in the process of feature recognition and similarity evaluation of work pieces, it is necessary to consider their process information such as surface roughness and tolerance of the main faces in addition to topological structure and shape size.
The traditional CAD model only records the topological structure, shape and geometry information, but cannot obtain the process information. Model Based Definition (MBD) is a model-based method for defining product data. Based on the topological structure and geometric data of the model itself, process information such as surface roughness, tolerances and annotations are added. Therefore, MBD based 3D CAD model similarity evaluation can not only solve the problem of model appearance similarity evaluation, but also can compare the process information of models, which expands the scope of information comparison and makes the model comparison result more accurate.

Based on the similarity of appearance and process, this paper proposes to address the problems of identifying similar local feature in target model and evaluating the similarity with source feature. The model from which local feature is searched is called target model, the model used for comparison is called source model, and the local feature used for comparison in source model is called source feature. First, the information set of the B-rep representation of source feature and target model is extracted, and then the MBD based Multi-Dimensional Attributed Adjacency Matrix (MDAAM) is constructed. Secondly, the faces of source feature and target model are taken as independent sets respectively, and the similarity between the faces of above two independent sets is taken as the weight to establish weighted complete bipartite graph. Thirdly, Kuhn-Munkres algorithm is used to find the optimal matching between two independent sets, and the adjacency between faces retrieved from the target model is determined. Finally, the similarity of matching faces between source feature and target model is evaluated.

The issues discussed in this article can be concisely described as follows. A face group is selected from source model as the source feature, and then the feature which is the most similar to the source feature is found from the target model as the local feature to be retrieved. At last, the similarity between source feature and local feature retrieved from the target model is calculated.

The remainder of the paper is organized as follows. Section 2 reviews the literature relating to feature recognition. In Section 3, method of information extraction from MBD based B-rep representative is introduced. In Section 4, MBD based MDAAM is constructed. In Section 5, MDAAM based AFR method and similarity evaluation method are proposed. The proposed methods are validated through a case study in Section 6, followed by conclusions in Section 7.

II. LITERATURE REVIEW

In order to identify the characteristics of different kinds of CAD models automatically, active research in the field of AFR has developed numerous techniques and systems, such as syntactic pattern recognition [2], artificial neural network [3], shape signatures [4], graph-based, rule-based, hybrid approach, etc. But some of the techniques have not been fully developed because of their limitations. For example, syntactic pattern recognition approach uses string based pattern recognition and natural language processing, with which it is difficult to express complex topological and geometrical structure of 3D models.

A. GRAPH-BASED FEATURE RECOGNITION

Graph-based feature recognition is one of the most studied methods [5]. It is difficult to identify features directly from the raw form elements directly extracted from the model. It is necessary to construct high-level semantic information as the basis of feature recognition by using the adjacency relationship between the faces in model and the attributes of each component element. So, in order to facilitate the recognition process, the concept of Attributed Adjacency Graph (AAG) built on the underlying B-rep is proposed by Joshi and Chang [6]. AAG is constructed by mapping model faces with vertices and mapping intersection lines between faces with edges, which is used to represent the topological structure and basic shape properties of the model. AAG is defined as a graph \( G = (N, A, T) \), where \( N \) is the set of nodes, \( A \) is the set of arcs and \( T \) is the set of attributes to arcs and the relationship between faces. The representation of AAG in computer can be a matrix, which is Attributed Adjacency Matrix (AAM) [7]. AAM is expressed as \( M = (C, R, A[C,R]) \), where \( C \) is the set of columns, \( R \) is the set of rows, and \( A[C, R] \) is the set of predicates of joint element in the matrix. \( C \) and \( R \) are mapped to faces and \( A[C, R] \) is mapped to joint element, which is used to express the relationship between faces.

Later, with the help of AAG and AAM, scholars have done a lot of work on feature recognition and similarity comparison. For example, Liu et al. [8] proposed a method to address the problem of extracting the Maximum Turnable State (MTS) with a AAG decomposition strategy. The feature recognition problem is handled by matching the feature shape descriptors with the descriptors of local element combinations out of the delta volume mesh. El-Mehalawi and Allen Miller [9], [10] proposed a kind of attributed graph to represent STEP format model, in which the nodes correspond to the surfaces of component, the links correspond to the edge and the attributes represent some primitive attributes of faces or edges. On the base of attributed graph, a method of retrieval and matching models is introduced by, which describes the topology of the model with nodes and arc, and depicts some limited shape information with attributes. Comparison between models is accomplished based on comparing the nodes’ surface type, the nodes’ number of edges, and on the edge compatibility.

In the aspect of model local identification and similarity evaluation, Zhang et al. [11] proposed a typical structure mining and similarity evaluation method of 3D model based on simulated annealing. By extracting the type of model edge, the angle between faces and the convexity of the edge, AAG and its association graph are constructed. The simulated annealing algorithm is used to detect the maximum clique in the association graph, so as to realize the mining and similarity comparison of typical structures. Liu et al. [12] proposed a hybrid feature recognition method based on graph
decomposition and feature factor clustering. By decomposing AAG into Minimum Attributed Adjacency Graph (MAAG), the difficulty of cross feature recognition is avoided. By matching MAAG with pre-defined feature elements, feature factors are obtained, and the feature factors satisfying certain rules are clustered into composite features, thus the machining features of cylinder parts can be effectively recognized. However, the pre-defined feature elements in this method are mainly for rotary features, so the recognition range is limited.

In the aspect of overall similarity comparison, Wang et al. [13] used weighted bipartite graph optimal matching method to evaluate the overall similarity of 3D CAD model. The model surfaces to be compared are divided into different classes, and each type of face is matched respectively. And then, the model similarity value is obtained by adding the optimal matching values of various types of faces. When calculating the similarity between faces, only basic parameters, such as face area, shape, side length, are considered, and the model cannot be evaluated comprehensively.

B. RULE-BASED FEATURE RECOGNITION

Rule-based feature recognition is another most studied method. According to the predefined rules that are characteristic to the feature, rule-based system uses algorithms to recognize features from models. Scholars have done a lot of work on it. For example, Oussama et al. [14] proposed a rule-based method to recognize rotational features. In order to development an automatic classification of product shape information, Zehtaban and Roller [15] utilized AFR approach to retrieve design models. A rule-based feature recognition method based on Opitz coding system was proposed for the automatic classification, and the final result of this model implies a predefined group of features. Campana and Mele [16] represented one rule-based approach of feature recognition for Design for Additive Manufacturing (DfAM). Geometrical entities are defined as graph-based representation, which is used for comparison with rules of DfAM to find the possible critical issues for manufacturability.

But for rule-based feature recognition method, it is necessary to formulate predefined rules for each class of features to be identified, and the definition of feature rules is not unique, and traditional rule-based feature recognition methods are computationally expensive. In addition, it is quite challenging for intersecting features [17].

C. HYBRID APPROACH FOR FEATURE RECOGNITION

Combining the advantages of single conventional feature recognition method, such as graph-based, rule-based, hint-based, schema, grammar, knowledge-based and so on, hybrid approach can overcome the limitations of single recognition method and there are many examples of successful applications.

For example, Guo et al. [18] and Sunil et al. [19] independently proposed the graph and rule-based hybrid 3D feature recognition for recognizing machining features. Based on graph-based feature recognition method, Li et al. [20], Gao et al. [21], Rameshbabu and Shunmugam [22] combined the methods of heuristic rules, hint-based and volume subtraction techniques respectively to propose hybrid approach for complex features.

Fougères and Ostrosi [23] adopted a hybrid schema, grammar and graph, to represent a feature. A grammar is defined by a 4-tuple to express knowledge and intelligence, and a topological and geometric entity graph is used to represent the topological and geometric relation of a feature. Feature recognition is carried out by multi-agent system and performs three stages: regioning, virtual extension and feature identification. Borkar and Puri [24] proposed a system of Feature Extraction (FE) for 3D components, which is a hybridization of volume decomposition and knowledge-based approach based on artificial intelligence planning. Li et al. [25] proposed a hybrid AFR method to recognize intersecting features with planar faces and quadric surfaces, which is a combination of hint-based, AAG and ANN.

D. OTHER FEATURE RECOGNITION METHODS

In addition to the above traditional feature recognition methods, many scholars have made other attempts and achieved satisfactory results.

For example, Zhang et al. [26] and Shi et al. [27] proposed the deep learning framework of feature recognition respectively. The former uses Deep 3D Convolutional Neural Networks (3D-CNNs) termed FeatureNet to learn machining feature from CAD models of mechanical parts, and the latter adopts Multiple Sectional View (MSV) representation for feature recognition. Gao et al. [28] and Wang [29] employed optimization algorithm, such as ant colony searching algorithm and wavelet transform of surface boundary, to search the optimal face matching sequence between the two models. Kim et al. [30] and Bespalov et al. [31] suggested volume decomposition methods for feature recognition. Kim et al. employed non-overlapping volume decomposition to overcome the problem of overlap of decomposed volumes and Bespalov et al. proposed a Scale-Space decomposition method to decompose 3D model into k sub-features to support matching and content-based retrieval of solid models.

Shi et al. [32] presented a feature recognition method using Heat Kernel Signature (HKS) for manufacturability analysis in Additive Manufacturing (AM). The proposed approach can identify geometric features and manufacturing constrains of different shapes. Al-wswasi and Ivanov [33] proposed a smart interactive automatic feature recognition methodology for recognizing the features of rotational parts, based on the extraction of the features’ geometrical and topological information. Woo and Kim [34] introduced one approach of protrusion feature recognition using a quantitative measure of orthogonal bounding factor (OBF). Yeo et al. [35] suggested a method of feature recognition to generate a measurement path for 5-axis On-machine measurement (OMM). Based on the surface to generate an optimal measurement path, the
coordinates are input at the position where the user desires to measure and the features are recognized.

E. DISCUSSION

The core of the above AFR method is to compare the predefined features or rules with target model to obtain the similar local features in target model. However, there is not one library of predefined features that can include all the possibilities of features, which could exist in a model. So, those methods are limited to the recognition of only a specific set of predefined features [1].

When compared with the predefined features in feature library, AAG only needs to express the adjacency relationship between faces and the angle relationship between faces, such as the convex or concave angle between two faces. According to the connection relationship between model faces, AAM is constructed and some graph theory related algorithms, such as subgraph isomorphism and optimal matching of complete bipartite graph, can be employed to compare the defined models in the model base and identify local features in target model.

Different from the above, in order to identify local feature from target model, it is necessary to define the face adjacency relationship of models, as well as other data such as shape, size, process information, etc. In traditional AAM, only topological information and primitive attribute information is recorded. When comparing source feature with target model, due to the lack of information, especially the process information such as tolerance and surface roughness, it is impossible to judge the process similarity of models. So, it is difficult to accurately identify the local features similar to the source features from target model and evaluate their similarity.

If there is not enough basic information, the model recognition will cause a deviation. As shown in Fig.1, the topological structures of the two hexahedrons in (a) and (b) are the same, but their shapes are quite different. However, according to the above method, when the traditional AAM is used to evaluate their similarity, their AAG and AAM are the same, as shown in (c) and Formula (28). So, the similarity of the two models is 1, but this is obviously not the case. Therefore, it is necessary to make a further comparison based on the more detailed model information to evaluate the similarity accurately.

\[
AAM = \begin{bmatrix}
  A & 1 & 0 & 1 & 1 & 1 \\
  1 & B & 1 & 0 & 1 & 1 \\
  0 & 1 & C & 1 & 1 & 1 \\
  1 & 0 & 1 & D & 1 & 1 \\
  1 & 1 & 1 & 1 & E & 0 \\
  1 & 1 & 1 & 1 & 0 & F \\
\end{bmatrix}
\]

The main purpose of AFR between two models is to compare their manufacturability and provide data basis and premise for Automatic Computer Aided Process Planning (ACAPP). The machining method is determined by the process information such as tolerance and surface roughness, which is also an important factor to be considered in the process of feature recognition. Even if the topological structure and geometric shape of models are identical, there may be significant differences in processing technology due to different surface machining accuracy. It can be seen that there is no systematic and complete method to accurately retrieve model feature, especially in the aspect of process similarity. Based on the above research, a new feature recognition and similarity evaluation method is established by considering the topological structure of model, the type and shape of elements such as face and edge, as well as the process information such as tolerance and surface roughness.

III. MBD BASED B-REP MODEL INFORMATION EXTRACTION

Automatic extraction of model information is not only the data basis of feature recognition, but also the key of accurate model similarity evaluation. The previous research on automatic feature recognition and model similarity inspection has done a lot of work in the extraction of key information such as model topological structure and element shape, but details such as dimension, tolerance and surface roughness of model faces cannot be extracted.

The expression of model is the basis of data extraction. In this paper, MBD based model is used as the data basis
of feature extraction and similarity judgment. As a model-based method for product lifecycle data definition, MBD adds tolerance, annotation and other process information on the basis of model topology and geometric data [36]. Therefore, the comparison and evaluation of process information can be added to feature recognition, which can improve the accuracy of feature recognition. At present, MBD has become a common model definition method. As early digital product definition standard, the definition rules of digital product were defined in ASME Y14.41, and then MBD product definition method was also defined in ISO 16792. In order to facilitate the standardized storage and reading of MBD data, STEP product digital standard is defined in ISO 10303 AP242 to manage model-based 3D engineering. According to the above product definition rules and product expression methods, most 3D software corporations have established data conversion interface for neutral format STEP, and also developed corresponding customized development tools, such as Creo toolkit of Creo software, UG Open provided by UG software, CATIA CAA of CATIA software, etc.

The extraction of model information in MBD is the foundation and key of automatic feature recognition. However, the existing 3D models are completed by different commercial CAD modelers. In terms of modeling method and data storage format, each type of software has its own proprietary data formats, which leads to the interoperability and CAD/CAM integration problems [37]. Therefore, it is difficult to find a general direct extraction method of 3D model information. Although different kinds of models have different data structures, their expression schema is basically stable, mainly including CSG and B-rep [38]. Model is expressed with many small cubes in CSG schema, with which it is not easy to express the model with complex shape, so CSG is less used in the current 3D model system [39]. B-rep schema has been dominant because it uniquely defines the faces and their patterns in a solid, which represents topological structure of models in the form of a relational model.

At present, most modeling software supports B-rep schema representation. In B-rep schema, models are represented with faces, which are bounded by sets of edges. A group of adjacent bounded surface elements is applied to express the boundary of models.

There are two methods to extract the information of MBD based B-rep model. The first is to extract information by reading the STEP neutral format file of the model [40], [41], and the second is to utilize customized development tools provided by commercial CAD modeler [42]. Here we use the latter to establish the method of automatic model data extraction, which is used to construct the data basis of model feature recognition and similarity comparison. In this paper, taking Creo as an example, through its customized development tool Creo Toolkit, we extract the information of topology, geometry, tolerance and surface roughness from model to provide data basis for feature recognition and similarity evaluation.

A. EXTRACTION OF MODEL TOPOLOGICAL STRUCTURE INFORMATION

Topological structure is a graph that represents the relationship between faces and lines. The topological structure information of a model represents the connection between the faces and edges, which expresses its basic shape. It is the primary content of feature recognition and similarity evaluation. The main functions involved in Creo Toolkit are shown in Table 1.

| No. | Name                                      | Function                                |
|-----|-------------------------------------------|-----------------------------------------|
| 1   | ProSolidSurfaceVisit()                    | Traverse the faces of a model and get handle array of faces |
| 2   | ProSurfaceContourVisit()                  | Traverse the contour of a face and get handle array of contour |
| 3   | ProContourEdgeVisit()                     | Traverse the edges of a contour and get handle array of edge |
| 4   | ProEdgeIdGet()                            | Get the ID of an edge through its handle |
| 5   | ProSurfaceIdGet()                         | Get the ID of a face through its handle |
| 6   | ProEdgeNeighborsGet()                     | Through ID of an edge, get the two IDs of its adjacent faces |
| 7   | ProSelect()                               | Select one or more faces of a model and get their handle |

B. EXTRACTION OF MODEL GEOMETRIC INFORMATION

The geometric information of MBD model mainly refers to the geometric shape, face area and line length of model. Here, the geometrical shape and area of faces, the shape and length of lines are extracted. The main functions involved in Creo Toolkit are shown in Table 2.

| No. | Name                                      | Function          |
|-----|-------------------------------------------|-------------------|
| 1   | ProEdgeTypeGet()                          | Get the type of an edge |
| 2   | ProEdgeLengthEval()                       | Get the length of an edge |
| 3   | ProSurfaceTypeGet()                       | Get the type of a face |
| 4   | ProSurfaceAreaEval()                      | Get the area of a face |

C. EXTRACTION OF TOLERANCE AND SURFACE ROUGHNESS

MBD model records the process information such as tolerance and surface roughness, which represents the process characteristics of the model. Here, size tolerance, form tolerance and surface roughness are extracted and processed. The main functions in Creo Toolkit are shown in Table 3.

IV. CONSTRUCTION OF MBD BASED MODEL REPRESENTING METHOD

MBD model contains not only topology and shape information, but also process information such as dimension, tolerance, annotation and etc. It is the premise of AFR and similarity evaluation to extract and express this information.
TABLE 3. Functions related to tolerance and an annotation.

| No. | Name                                      | Function                  |
|-----|-------------------------------------------|---------------------------|
| 1   | ProAnnotationElementGetType()             | Get the type of an annotation |
| 2   | ProSolidSurfinishVisit()                  | Traverse the surface roughness of a model |
| 3   | ProSurfinishValueGet()                    | Get the value of a surface roughness |
| 4   | ProSurfinishReferencesGet()               | Get the reference of a surface roughness |
| 5   | ProMdlNoteVisit()                         | Get all the annotation of a model |
| 6   | ProNoteTextGet()                          | Get the note text of a model |
| 7   | ProSolidAnnotationElementsVisit()         | Traverse all the annotations of a model |
| 8   | ProAnnotationElementAnnotationGet()       | Get the shape and positional tolerance of a model |
| 9   | ProGtoldataTypeGet()                      | Get the type of a tolerance |
| 10  | ProGtoldataValueGet()                     | Get the value of a tolerance |
| 11  | ProAnnotationElementReferencesCollect()    | Get the reference of a tolerance |

in multidimensional and multi-level form. In order to express the MBD based information comprehensively and clearly, based on AAG and AAM, MDAAM is proposed to represent MBD based model, which describes the topology, geometry, process and other information of the model with multiple matrices. The MDAAM provides data basis for later AFR and similarity evaluation, as shown in Fig. 2.

In the preceding MDAAM, TM is the geometric carrier of the model, and its diagonal elements express faces and non-diagonal elements express the lines between surfaces. On the base of TM, SM and GDM represents the shape of faces and lines. Their diagonal elements stand for the shape and area of the corresponding face in TM, and the non-diagonal elements express the shape and length of the model lines. SRM and FTM express the face roughness and shape tolerance, and the corresponding tolerance values are expressed by its diagonal elements.

Before feature recognition, first extract the model information of faces and lines to construct TM, and then, according to the ID in TM, traverse their shape parameters, roughness and shape tolerance to construct MDAAM, which provide data basis for later AFR and similarity evaluation.

A. ESTABLISHMENT OF TM

The topological structure of model represents the connection relationship between faces and their intersection lines, which is the most important reference for model AFR and similarity evaluation. B-rep schema of a model records its detailed information about topological structure, so model topology can be established by extracting the information from its B-rep schema. With the help of graph theory, model faces are represented by vertices of the graph, and the intersection lines between faces are represented by connecting edges between vertices. Together with the attribute value reflecting the shape of face or line, AAG is constructed to represent the topological structure of model. In order to facilitate the use of model data, AAM is established by taking the vertices in AAG as diagonal elements and the edges between vertices as non-diagonal elements.
In the process of constructing AAG, the connection edges between model faces should be processed first. The faces of industrial products mainly include planar plane, revolving surface, cylindrical surface and other types of surfaces. In the B-rep schema of a model, a revolution or cylinder is composed of two parts, as shown in Fig. 3 (a) and (b). When extracting its information, it will be dealt as two faces [43]. According to the preceding information reading method from B-rep schema model, the AAG of the above two models is shown in Fig. 3 (c). However, the vertices connected by multiple edges, which ensures the uniqueness of intersection lines between faces. When faces do not intersect, the value is assigned to zero.

FIGURE 3. Revolution, cylinder and their AAG.

In terms of the surface formation method, for the revolving surface shown in Fig. (a) and the ruled surface shown in Fig. (b), the intersection lines of the two half surfaces in B-rep schema are its generatrix on model surface. Therefore, the shape and size of two intersection lines in the surface of revolution or cylinder are exactly the same. For the convenience of representation and operation, the two intersection lines are simplified to one edge. The simplified AAG is shown in Fig. 3(d). After the preceding simplification, the AAG of the model with cylinder or revolution will no longer include multiple edges, which ensures the uniqueness of intersection lines of the two surfaces. The AAM established from Fig. 3 is shown in Formula (29), which is an upper triangular matrix.

$$\text{AAM} = \begin{bmatrix} A & 0 & 1 & 2 \\ B & 3 & 4 \\ C & 5 \\ D & \end{bmatrix}$$ (2)

According to the preceding method of establishing AAM, the first dimension matrix of MDAAM is established as shown in Formula (13), which is used to represent the topological structure of the model.

$$\text{TM} = \begin{bmatrix} FT_{11} & ET_{12} & \cdots & ET_{1i} & \cdots & ET_{1n} \\ FT_{22} & ET_{2i} & \cdots & ET_{2i} & \cdots & ET_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ FT_{ii} & ET_{ii} & \cdots & ET_{ii} & \cdots & ET_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ FT_{nn} & \end{bmatrix}$$ (3)

where, the diagonal elements in the matrix $FT_{ii}$ ($i = 1, 2, \ldots, n$) represent the ID of model faces. Other elements $ET_{ij}$ ($i, j = 1, 2, \ldots, n$) represent the ID of intersection lines between faces. When faces do not intersect, the value is assigned to zero.

The above TM can be completed by traversing the faces and edges of model, and then retrieving the adjacent faces of model edges. Its pseudo code is as shown Algorithm 1.

**B. ESTABLISHMENT OF SM**

The elements in $SM$ and those in the same position in $TM$ represent the same element, which represents the shapes of the faces and edges in model. $SM$ is the second dimension matrix of MDAAM, which is also an $n$-order square matrix. $n$ represents the number of faces in the model. $SM$ is as follows,

$$\text{SM} = \begin{bmatrix} SS_{11} & ES_{12} & \cdots & ES_{1i} & \cdots & ES_{1n} \\ SS_{22} & ES_{2i} & \cdots & ES_{2i} & \cdots & ES_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ SS_{ii} & ES_{in} & \cdots & ES_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ SS_{nn} & \end{bmatrix}$$ (4)

where, the diagonal elements in the matrix $SS_{ii}$ ($i = 1, 2, \ldots, n$) represent the faces of the model, and different values are assigned according to different types of faces. The definition is shown in Formula (5). Other elements represent the intersection lines between faces. When faces do not intersect, the value is assigned to zero. When face intersects with other one, different values are assigned according to the type of intersection line, as shown in Formula (6).

$$SS_{ii} = \begin{cases} 1 & \text{plane} \\ 2 & \text{cylindrical} \\ 3 & \text{cone} \\ 4 & \text{torus} \\ 5 & \text{spline} \\ 6 & \text{others} \end{cases}$$ (5)
Algorithm 1 The Construction of TM

1. Input: MBD based model
2. Output: TM of the model
3. The information of faces and edges is extracted and written into array $F[i]$ in $TM[i][j]$.
4. surface[] ← ProSolidSurfaceVisit() //Traverse all the faces of the model
5. $nbr$ ← ProArraySizeGet() //Get the number of faces in the model
6. define $F[nbr]$ //Store the information of model faces. $F[i]$ ← ID of No. ($i + 1$) face
7. define $E[m][3]$ //Store the information of model edges. $m$ is big enough and stands for the number of edges. $E[i][0]$ ← ID of No. ($i + 1$) edge; $E[i][1]$ ← ID of the first neighbor face of No. ($i + 1$) edge; $E[i][2]$ ← ID of the second neighbor face of No. ($i + 1$) edge
8. for ($i = 0; i < nbr; i++$) do //Traverse all the faces of the model
9. $F[i]$ ← ProSurfaceContourVisit() //Get the ID of No. ($i + 1$) face
10. contour[] ← ProSurfaceContourVisit() //Traverse the contour of current face.
11. Output: array of contour
12. for ($j = 0; j < c; j++$) do //Read all the edges from current contour
13. edge[] ← ProContourEdgeVisit() //Traverse all the edges from current contour.
14. Output: array of edges
15. $e$ ← ProArraySizeGet() //Get the number of edge in edge[]
16. for ($k = 0; k < e; k++$) do //Read the information of edge from its handle
17. EID ← ProEdgeGet() //Get the ID of current edge
18. NSID1 and NSID2 ← ProEdgeNeighborsGet() //Get the ID of neighbor faces
19. cursnbr ← the number of edges in current $E[n][3]$.
20. $y$ ← 0
21. for ($x = 0; x < cursnbr; x++)$ do //Judge if the current edge is written or multi-edge
22. if ($EID == E[x][0]$) then //The current edge has already been written
23. break
24. end if
25. $y++$
26. end for
27. if ($y == cursnbr$) then //Write the edge
28. $E[0][0]$, $E[1][1]$, and $E[2][2]$ ← EID, NSID1 and NSID2.
29. end if
30. end for
31. end for
32. for ($k = 0; k < e; k++$) do //Read the information of edges and write them to TM
33. $e$ ← ProArraySizeGet() //Get the number of edge in edge[]
34. if ($((NSID1 == E[x][2] & NSID2 == E[x][3]) || (NSID1 == E[x][3] & NSID2 == E[x][2]))$) then //multi-edge
35. break
36. end if
37. $y++$
38. end for
39. if ($y == cursnbr$) then //Write the edge
40. $E[0][0]$, $E[1][1]$, and $E[2][2]$ ← EID, NSID1 and NSID2.
41. end if
42. end for
43. end for
44. Construct TM according the preceding information
45. define TM[nbr][nbr] and its initial value is 0s/
46. for ($i = 0; i < nbr; i++$) do //Write the information of faces to TM
47. TM[i][i] ← F[i][0] //Store the ID of No. ($i + 1$) face to TM[i][i]
48. end for
49. for ($u = 0; u < nbr; u++$) do //Read the information of edges and write them to TM
50. if ($F[u] == E[i][1] || (F[u] == E[i][2])$) then //The ID of No. ($u + 1$) face is the same as the ID of one neighbor face of No. ($i + 1$) edge
51. face[] ← ProContourEdgeVisit() //Traverse all the edges from current contour.
52. Output: array of edges
53. end if
54. end for
55. end for

where, the diagonal elements in the matrix $SGD_i$ ($i = 1, 2, ..., n$) represent the area of the faces, which can be obtained by using the customized development function of software. For example, for a model created by Creo, the area of a face can be obtained with the help of ProEdgeLengthEval() function. Non diagonal elements $EGD_{ij}$ ($i, j = 1, 2, ..., n$) represent the length of intersection lines between faces. If two faces do not intersect, the element value is 0. For the model created by Creo, the length of a line can be obtained with the help of ProEdgeLengthEval() function.

D. ESTABLISHMENT OF SRM

The roughness of machined face reflects the machining quality of mechanical parts, which is an important symbol of its quality grade. For the faces with different roughness, their processing technology are different, and different machine tools and processing methods will be used in process planning. SRX records the roughness of the mating faces and other important surfaces, so the non diagonal elements are zero, and it is a diagonal matrix, shown as follows,

$$SRM = \begin{bmatrix} SR_{11} & \cdots & SR_{1i} \\ \vdots & \ddots & \vdots \\ SR_{ij} & \cdots & SR_{nn} \end{bmatrix}$$

For the MBD model created by Creo, the two functions of ProSurfaceTypeGet() and ProEdgeTypeGet() are used to obtain the types of faces and edges respectively, and then the formulas (5) and (6) are used to convert them into integer. At last, write the integer to SM in the same position as TM.
roughness array is retrieved with the help of ProSolidSurfinishVisit() function, and then obtain the value of roughness using ProSurfinishValueGet() function and get the face reference where the roughness lies by ProSurfinishReferencesGet() function.

**E. ESTABLISHMENT OF STM**

Size tolerance reflects the machining accuracy of parts and the matching properties of assemblies. For parts with different tolerance grades, the processing complexity and cost are also different. Therefore, size tolerance is also a factor to be considered in the process of AFR and similarity evaluation. For most models, the assembly tolerance between hole and shaft is the key factor, so here we mainly consider the tolerance grade and deviation of hole and shaft. Such tolerance only exists on rotary surface, so STM is a diagonal matrix, which is defined as follows,

$$STM = \begin{bmatrix}
ST_{11} & & & & & \\
& ST_{22} & & & & \\
& & \ddots & & & \\
& & & \ddots & & \\
& & & & ST_{ii} & \\
& & & & & \vdots \\
& & & & & ST_{nn}
\end{bmatrix}$$

(9)

where, elements $ST_{ii}$ ($i = 1, 2, \ldots, n$) represent the size tolerance of revolving surface, which are character data and consist of four characters. The first two are letters, indicating fundamental deviation, and the last two are numbers, indicating tolerance grade. Its value is generally between 05-12, which is consistent with the tolerance grade of common hole and shaft fit. For example, the size tolerance of φ100H7 is expressed as 0H07. If there is no tolerance requirement on the surface, the value is 0000.

**F. ESTABLISHMENT OF FTMG**

Form tolerance determines the main machining methods of part surface, and it is also one of the references for AFR and similarity evaluation. Here six types of form tolerance is utilized: straightness, flatness, circularity, cylindricity, profile of a line and profile of a surface. All of these tolerances are applied to model faces, so the matrix is diagonal. In the actual modeling process, some faces may have more than one kind of shape tolerance based on process requirements. Therefore, each element of the shape tolerance matrix is defined by a six-dimensional array, which is called FTMG. Its definition is as follows,

$$FTMG = \begin{bmatrix}
FT_{11}(6) & & & & & \\
& FT_{22}(6) & & & & \\
& & \ddots & & & \\
& & & \ddots & & \\
& & & & FT_{ii}(6) & \\
& & & & & \vdots \\
& & & & & FT_{nn}(6)
\end{bmatrix}$$

(10)

where, the elements $FT_{ii}(6)$ ($i = 1, 2, \ldots, n$) represent the form tolerance array of model face, and 6 is the dimension of the array. Array elements consist of form tolerance types and their values, which are defined as follows,

$$FT_{ii}(j) = \begin{cases}
ST & \text{straightness} \\
FL & \text{flatness} \\
CI & \text{circularity} \\
CY & \text{cylindricity} \\
PL & \text{profile of a line} \\
PS & \text{profile of a surface} \\
0 & \text{no form tolerance}
\end{cases}$$

(11)

where, ST, FL, CI, CY, PL and PS stand for tolerance types, and ** are tolerance values.

**V. MDAAM BASED MODEL LOCAL FEATURE RECOGNITION AND SIMILARITY EVALUATION**

On the basis of the above mentioned MDAAM information set, a two-stage local feature recognition and similarity evaluation method combining optimal matching and adjacency judgment is proposed. Bipartite graph is constructed based on the independent sets formed from the faces of source feature and target model. And then, based on MDAAM, the similarity of each pair of vertices between source feature and target model is calculated, which is used as the weight of the connecting edge of bipartite. With the preceding independent sets and weight, weighted bipartite graph is constructed. The Kuhn-Munkres algorithm is utilized to find the optimal matching between source feature and a local feature of the target model, in which the sum of matching degree is the highest among all matching schemes. Then, adjacency judgment is made for the optimal matching local face set of target model, so as to exclude isolated faces and ensure the continuity of the retrieved faces. Finally, the similarity of the optimal matching faces between the local feature of target model and the source feature is calculated to complete the similarity evaluation.

In this paper, the following rules are formulated for the evaluation of face similarity between models:

**Rule 1:** The similarity evaluation between models includes appearance evaluation and manufacturability evaluation. The former refers to the evaluation of the similarity of topological structure, shape of face and edge, and all kinds of sizes between models. The latter refers to the evaluation of the similarity of process information such as machining tolerance, surface roughness and etc.

**Rule 2:** If the types of the matching faces are different, the similarity between the two faces is considered to be zero. Only faces of the same type have non-zero similarity coefficients.

**A. CONSTRUCTION OF WEIGHTED COMPLETE BIPARTITE GRAPH**

In order to express the similarity between the faces of source feature and target model, a bipartite graph $G$ is established
with the faces of the two models as independent sets and the path from the face of source feature to the face of target model as edge, which is shown as follows,

$$G = (V_S, V_T, E)$$  \hspace{1cm} (12)$$

where, $V_S$ is the independent set established with the source feature faces as vertices, $V_T$ is the independent set established with the target model faces as vertices, and $E$ is the edge set between $V_S$ and $V_T$.

In general, the number of vertices in source feature is less than that in target model. Therefore, it is necessary to add supplementary faces to the independent set of source feature, so that the number of vertices in the two independent sets is equal.

Complete bipartite graph is constructed by adding connecting edges established from each face in the independent set of source feature to all the faces in independent set of target model. If the number of supplementary faces added is $h$, then there is,

$$|V_S| + h = |V_T|$$  \hspace{1cm} (13)$$

where, $|V_S|$ is the number of vertices in the independent set of source feature, that is, the number of source feature faces. $|V_T|$ is the number of vertices in the independent set of target model.

The weight of each edge in the bipartite graph is calculated based on the face similarity between source feature independent set and target model independent set. When doing this, the main factors considered include the type similarity $TY$, shape similarity $SH$, surface roughness similarity $SR$, size tolerance similarity $ST$ and form tolerance similarity $FT$ between source feature and target model. In order to ensure the practical application significance of similarity matching, the type similarity between matching faces is the first factor to be considered. According to preceding rule 2, if the types of faces do not match, the similarity between matching faces is regarded as zero.

According to the preceding rules, the similarity weight between faces is divided into two parts: type weight and similarity weight. The calculation formula is as follows,

$$\omega_{(S,T)} = \omega_{TY} \times \omega_{Sim}$$  \hspace{1cm} (14)$$

where, $\omega_{(S,T)}$ is the weight of a path from a face of source feature to a face of target model. $\omega_{TY}$ is type weight, which represents the type weight of the matching faces, and its value is shown in formula (15). $\omega_{Sim}$ is the similarity weight, which represents the similarity weight of the matching surface, and its definition is shown in Formula (16)

$$\omega_{TY} = \begin{cases} 1 & \text{When face type is the same} \\ 0 & \text{When face type is different} \end{cases}$$  \hspace{1cm} (15)$$

$$\omega_{Sim} = A \times \omega_{SH} + B \times \omega_{SR} + C \times \omega_{ST} + D \times \omega_{FT}$$  \hspace{1cm} (16)$$

where, $\omega_{SH}$, $\omega_{SR}$, $\omega_{ST}$ and $\omega_{FT}$ represent shape similarity, surface roughness similarity, size tolerance similarity and form tolerance similarity between matching surfaces, and constants A, B, C and D represent the weight coefficients of the above four similarities respectively. According to the different research purposes, the weight coefficients take different values. As shown in Table 4, when evaluating the similarity of local features of the model, if only appearance evaluation is carried out, the weight of A is taken as 1, and the weights of B, C and D are taken as zero, which indicates that the similarity is only affected by appearance. When process similarity evaluation is conducted, each weight is taken as the average value, which indicates that appearance, surface roughness, size tolerance and form tolerance play an equally important role in similarity evaluation.

| TABLE 4. The value of weight coefficient. |
|-----------------------------------------|
| weight coefficient | A | B | C | D |
|---------------------|---|---|---|---|
| Appearance evaluation| 1 | 0 | 0 | 0 |
| process evaluation  | 0.25 | 0.25 | 0.25 | 0.25 |

Literature [44] introduces a normalized measure to detect the difference between two values. The difference between two variables $A$ and $B$ is shown as follows,

$$d(A, B) = \frac{|A - B|}{\max(A, B)}$$  \hspace{1cm} (17)$$

where, the result of $d(A, B)$ is a value between 0 and 1, indicating the difference between A and B.

In order to measure the similarity between the two elements, based on the preceding Formula (17), construct the following formula,

$$s(A, B) = 1 - d(A, B)$$  \hspace{1cm} (18)$$

where, $s(A, B)$ represents the degree of similarity between A and B, with values between 0-1. The more similar A is to B, the closer the value is to 1. When A is equal to B, its value is 1.

According to the above elements similarity evaluation method, based on the attribute information extracted from model, the appearance similarity $\omega_{SH}$ between models is constructed as follows,

$$\omega_{SH} = \left(1 - \frac{|A_S - A_T|}{\max(A_S, A_T)}\right) \times \left(1 - \frac{|C_S - C_T|}{\max(C_S, C_T)}\right) \times \left(1 - \frac{|N_S - N_T|}{\max(N_S, N_T)}\right)$$  \hspace{1cm} (19)$$

where, $A_S$ and $A_T$, $C_S$ and $C_T$, $N_S$ and $N_T$ represent the area, perimeter, and number of edges of the matching surface between source feature and target model, respectively. The area of each surface is read directly from the diagonal elements of GDM $SGD_{ii}$ ($i = 1, 2, \ldots, n$) shown in Formula (4). The perimeter of No.i face is the sum of values of the non diagonal elements in the No.i row of GDM. The number of edges of No.i face is the number of non diagonal elements with non-zero values in No.i row of GDM.

Surface roughness and size tolerance are important process information and the basis for the similarity judgment.
of model processing technology, which directly determines the machining equipment such as machine tools, fixtures and other processing equipment used in the product processing process. The similarity calculation of surface roughness and size tolerance are shown in Formula (20) - (21) respectively.

\[
\omega_{SR} = \begin{cases} 
1 - \frac{|SR_S - SR_T|}{\max (SR_S, SR_T)} & \text{Both faces have roughness} \\
1 & \text{None of the faces has roughness} \\
0 & \text{One of the faces has roughness}
\end{cases} \tag{20}
\]

\[
\omega_{ST} = \begin{cases} 
1 - \frac{|ST_S - ST_T|}{\max (ST_S, ST_T)} & \text{Both faces have size tolerance} \\
1 & \text{None of the faces has size tolerance} \\
0 & \text{One of the faces has size tolerance}
\end{cases} \tag{21}
\]

In Formula (20), \(\omega_{SR}\) is the similarity value of surface roughness between faces. \(SR_S\) and \(SR_T\) are the surface roughness of matching faces in source feature and target model respectively. In Formula (21), \(\omega_{ST}\) is the similarity value of size tolerance between faces. \(ST_S\) and \(ST_T\) are the size tolerance of matching faces in two models respectively.

Form tolerance is another important manufacturability parameter of a part, and a face may contain several form tolerances. According to the six kinds of form tolerances commonly used in the process of mechanical part design, its similarity calculation formula is constructed as follows,

\[
\omega_{FT} = \frac{1}{h} \times \sum_{i=1}^{h} [1 - d(FT_{Si}, FT_{Ti})] \tag{22}
\]

where, \(\omega_{FT}\) represents the value of form tolerance similarity between two faces, \(h\) represents the number of form tolerance contained in the matching face, \(h \leq 6\), and \(i\) represents the serial number of form tolerance. \(d(FT_{Si}, FT_{Ti})\) represents the difference between the two values of the No. \(i\) form tolerance, and its definition is (23), as shown at the bottom of the page, where, \(FT_{Si}\) and \(FT_{Ti}\) stand for the value of the No. \(i\) form tolerance in source feature and target model respectively.

According to the definition of bipartite graph in Formula (29), as well as the preceding definition of edge weights on model faces, a weighted complete bipartite graph is established, as shown in Fig. 4. In the figure, the source feature contains \(m\) faces and the target model contains \(n\) faces. In order to ensure the equal number of the two model faces, \(h\) supplementary faces are added to the source feature. Taking the faces of source feature and target model as independent sets, connecting edges are established between each pair of vertices in the above two independent. And then the similarity of the two faces connected by the edges is taken as the weight, and weighted complete bipartite graph is finally constructed.

In order to facilitate the later calculation, the weights of the connecting edges in the bipartite graph are extracted to construct a weight matrix, as shown in Formula (24).
The similarity between the faces of source feature and target model is expressed in the form of matrix elements.

\[
WM = \begin{bmatrix}
\omega(1,1) & \omega(1,2) & \cdots & \omega(1,j) & \cdots & \omega(1,n) \\
\omega(2,1) & \omega(2,2) & \cdots & \omega(2,j) & \cdots & \omega(2,n) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\omega(i,1) & \omega(i,2) & \cdots & \omega(i,j) & \cdots & \omega(i,n) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\omega(m,1) & \omega(m,2) & \cdots & \omega(m,j) & \cdots & \omega(m,n) \\
\omega(m+1,1) & \omega(m+1,2) & \cdots & \omega(m+1,j) & \cdots & \omega(m+1,n) \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\omega(m+h,1) & \omega(m+h,2) & \cdots & \omega(m+h,j) & \cdots & \omega(m+h,n)
\end{bmatrix}
\]  
(24)

where, \( \omega(i,j) \) is the weight between the No. \( i \) face of source feature and the No. \( j \) face of target model, representing the similarity between the two faces; \( m \) is the number of source feature faces; \( m = |V_S| \), \( n \) is the number of target model faces, \( n = |V_T| \), \( h \) is the number of supplementary faces added to source feature independent set \( V_S \), in order to ensure that the number of faces in the two independent sets \( V_S \) and \( V_T \) is equal.

**B. THE OPTIMAL MATCHING BETWEEN THE FACES OF SOURCE FEATURE AND TARGET MODEL**

On the basis of the above-mentioned weighted complete bipartite graph, Kuhn-Munkres algorithm [45] is used to search the optimal matching \( M \) between the independent sets of bipartite graph. After removing the matching of supplementary faces with zero weight, the local feature most similar to source feature in the target model is obtained.

Using Kuhn-Munkres algorithm to solve the optimal matching problem of bipartite graph is completed by introducing the method of feasible vertex labeling (f. v. l.), which transforms the optimal matching problem of bipartite graph into the maximum weighted matching under the perfect matching of bipartite graph. The steps of solving the optimal matching of weighted complete bipartite graph \( G=(V_S, V_T, E(s,t)) \) are as follows.

1. The f. v. l. of each vertex is determined. Here the initial values are as follows,

\[
\begin{align*}
 l(s) &= \max \{ \omega(s,t) \} \quad s \in V_S \\
 l(t) &= 0 \quad t \in V_T
\end{align*}
\]  
(25)

where, \( s \) and \( t \) represent one face in source feature and target model respectively, which are represented by a vertex in both independent sets. \( l(s) \) and \( l(t) \) are the initial values of f. v. l. of \( s \) and \( t \), respectively. \( V_S \) and \( V_T \) are the two independent sets of bipartite graphs, respectively. \( \omega(s,t) \) is the weight of \( s \) and \( t \). \( l(s) \) and \( l(t) \) satisfy the condition: \( l(s) + l(t) \geq \omega(s,t) \), \( \forall s \in V_S, t \in V_T \).

2. Determine the equality subgraph \( G_l \), and take any matching \( M \) in \( G_l \) as the initial matching. The equal subgraph is a spanning subgraph of \( G \) with edge set \( E_l \), which satisfies the condition: \( E_l = \{(s, t) \in E | l(s) + l(t) = \omega(s,t)\} \).

(3) Judgment: if \( M \) saturates each vertex of \( V_S \), then \( M \) is the optimal matching, and the optimal matching result is output; Otherwise, any unsaturated vertex \( u \) of \( M \) is selected in \( V_S \), and the vertex set \( S \) is made, which is the set of vertex \( u \). The vertex set \( W \) is established: \( W \leftarrow \emptyset \).

(4) If \( N_G(S) = W \), update the f. v. l. of the two independent sets \( V_S \) and \( V_T \), and their values are shown in Formulas (26) - (27). After updating, replace \( l \) with \( l' \) and \( G_l \) with \( G_{l'} \).

\[
a_t = \min \{ l(s) + l(t) - \omega(s,t) \} \quad s \in S, \ t \notin W
\]  
(26)

where, \( s \in S, t \notin W \) indicates that \( s \) is the matched point of independent set \( V_S \), and \( t \) is the unmatched point of independent set \( V_T \).

\[
l' (v) = \begin{cases} 
 l(v) - a_t & v \in V_S \\
 l(v) + a_t & v \in V_T \\
 l(v) & \text{other}
\end{cases}
\]  
(27)
where, only the f. v. l. of the matched points of independent set \( V_S \) and \( V_T \) are updated.

(5) Select \( t \in N_{G_i}(S) \setminus W \), that is, the selected \( s \) belongs to set \( N_{G_i}(S) \) but not to set \( W \). If \( t \) is \( M \)-saturated, then \( t \) has been matched. Let the match of \( t \) in \( M \) be (t, s), add this \( t \) to set \( W \), add \( s \) matching \( t \) to set \( s \), and go to (4). If \( t \) is \( M \)-unsaturated, then \( t \) can be used as the starting point of the augmenting path. Let \( P \) be the \( M \)-augmenting (u, t) path \( M \leftarrow M \Delta E \) (P) in \( G_i \), then turn to (3).

Through the preceding algorithm, the matching \( M \) is obtained and its weight \( \omega(M) = \sum_{e \in M} \omega(e) \) is maximum, which is the correspondence between the faces of target model and source feature.

The pseudo code of Kuhn-Munkres algorithm is as shown Algorithm 2.

**Algorithm 2 Kuhn-Munkres Algorithm**

1. **Input**: weighted complete bipartite graph \( G = (V_S, V_T, E) \) and its weight matrix \( WM \), represented by \( sim[n][n] \), \( n \) is the number of faces in independent set
2. **Output**: optimal matching \( M[n] \) between \( V_S \) and \( V_T \), \( M[n] \) represents the serial number of the face in \( V_T \) that matches the No.\( i \) face in \( V_S \)
3. \( \omega \): The definition of variables/s
4. define INF as large enough integer constant
5. define \( exS[n] \) and \( exT[n] \):Represent the f. v. l. of the vertices in \( V_S \) and \( V_T \)
6. define \( visS[n] \) and \( visT[n] \):Record whether vertices of \( V_S \) and \( V_T \) are matched in each round of matching
7. define \( M[n] \): Record the vertex in \( V_T \) that matches the No.\( i \) vertex in \( V_S \)
8. if there is no match, it is -1
9. define \( slack[n] \): Record the minimum value that the f. v. l. needs to change if the vertex in \( V_T \) can be matched by the vertex in \( V_S \)
10. initialize \( match[n] \) with initial value -1 //At the beginning, every vertex in \( V_T \) has no matched vertex in \( V_S \)
11. initialize \( exT[n] \) with initial value 0
12. \( for \): Do the f. v. l. \( 0 < i < n \), \( i = i + 1 \) do //Initializes the f. v. l. of No.i vertex is the maximum \( \omega_{\text{MAX}} \) of the No.i row in \( WM \)
13. \( exS[i] \leftarrow \) the maximum value of No.i row in \( sim[n][n] \)
14. \( end for \)
15. initializes \( slack[] \) with initial value INF
16. \( while \): //Try to find matching vertex. If not found, decrease f. v. l. until found
17. initializes \( visS[] \) and \( visT[] \) with initial value FALSE://Record whether vertices have been tried to match
18. if \( \text{search_path}(i) \) then // Call search_path(). If succeed, quit
19. break
20. end if
21. \( // \) If not found, decrease f. v. l.
22. int \( d \) = INF
23. \( for \) \( (int j = 0; j < n; j = j + 1) \) do
24. if \( (\text{visT}[j]) \) then
25. \( d \leftarrow \min(d, \text{slack}[j]); \)
26. end if
27. \( end for \)
28. \( for \) \( (int j = 0; j < n; j = j + 1) \) do
29. if \( \text{visS}[j] \) then //Decrease all the f. v. l. of visited vertices in \( V_S \)
30. \( exS[j] \leftarrow d \)
31. end if
32. if \( \text{visT}[j] \) then //Increase all the f. v. l. of visited vertices in \( V_T \)
33. \( exT[j] \leftarrow +d \)
34. else
35. \( \text{slack}[j] \leftarrow d \)
36. end if
37. \( end for \)
38. \( end while \)
39. \( end for \)
40. bool \( \text{search_path}(int VS) \): VS represent vertex in \( V_S \)
41. { \n42. \( visS[VS] = true \)
43. \( \text{for}(\int VT = 0; VT < n; VT = \ldots) \) do
44. \( \text{if}(\text{visT}[VT]) \) then
45. continue
46. end if
47. \( \text{int gap} = \text{exS[VS]} + \text{exT[VT]} - \text{sim[VS][VT]} \);
48. \( \text{if}(\text{gap} = 0) \) then
49. \( \text{visT[VT]} = true \)
50. \( \text{if}(\text{M}[VT] = 0 \oplus \text{search_path}(M[VT])) \) then //VT in \( V_T \) has no matching or the matching vertex in \( V_S \) of \( V_T \) has other matching
51. \( \text{M}[VT] = VS \)
52. return true
53. end if
54. else
55. \( \text{slack}[VT] = \min(\text{slack}[VT], \text{gap}) \)
56. end if
57. \( end for \)
58. return false
59. }
(but not sufficient) condition is that any one of its faces is adjacent to at least one other face, that is, in AAG of solid continuous face group, the degree of any vertex is greater than or equal to 1.

Use the method of proof to the contrary to prove the preceding criteria. If face A belongs to a solid and is not adjacent to other faces in face group, then face A is an isolated surface on the boundary of this solid. In this case, the solid model cannot be closed and the solid cannot be generated in B-rep schema. Therefore, it can be concluded that the face on a face set is adjacent to at least one face in this face set.

Therefore, if a face is not adjacent to other optimal matching faces in target model face set T, it does not belong to
FIGURE 13. Correspondence between the faces of source feature and target model.

FIGURE 14. Correspondence between the faces of source feature and target model after weight reduced.

FIGURE 15. Source model.

FIGURE 16. Target model.

this set and is an isolated face, and the optimal matching is not the most similar local feature of target model. Here, the detection of isolated face is realized by the method of retrieving whether one face contains the edge shared with other faces in the set. Its flow chart is shown in Fig. 6, and its pseudo code is shown Algorithm 3.
For the isolated surface in optimal matching, we propose a method of rematch by reducing the weight between the isolated face and the face in source feature. Its procedure is as follows.

1) REDUCE WEIGHT
Reduce the weight between the isolated face in target model and the surface in source feature. The new weights are shown below,

$$w'(S(i), T(isosurf)) = w(S(i), T(isosurf)) \times (1 - \alpha)$$  \hspace{1cm} (28)

where, $w'(S(i), T(isosurf))$ is the new weight between the isolated face in target model and the faces in source feature. $w(S(i), T(isosurf))$ is the original weight. $i$ is the serial number of source feature, $i = 1, 2, \ldots, m$. $\alpha$ is the ratio of weight reduction each time, and the default value is 10%.

---

**FIGURE 17.** Basic surface information of source feature.

**FIGURE 18.** Basic edge information of source feature.

**FIGURE 19.** Basic surface information of target model.
2) RECALCULATE THE OPTIMAL MATCHING
After the above weight reduction, \( \omega'(S(i), T(\text{isosurf})) \) is used to replace \( \omega(S(i), T(\text{isosurf})) \) to recalculate the weight matrix shown in Formula (23), and the optimal matching is recalculated according to Kuhn-Munkres algorithm in 5.2.

3) REMOVE SUPPLEMENTARY SURFACE
Remove the matching faces with supplementary faces of source feature in target model. Face set T is obtained.

4) ADJACENCY JUDGMENT
According to the preceding method in this section, the adjacency judgment of the local optimal matching face of target model is conducted again, and the isolated face is processed until all the faces are adjacent faces.

According to the above algorithm, the pseudo code of the optimal matching face adjacency judgment and isolated face processing flow is as shown Algorithm 4.

D. LOCAL FEATURE SIMILARITY EVALUATION
By judging the adjacency between the matching faces, isolated face which do not belong to the local features are excluded, and the face set matching source features in target model is obtained. According to the different matching weights, the matching face pairs can be divided into three cases.

1) THE MATCHING WEIGHT IS 1
In this kind of face pair, the matching face is completely the same as the source feature face in every aspect of shape and manufacturability, such as type, size, tolerance, surface roughness, and the similarity between them is 1.

2) THE MATCHING WEIGHT IS \( 0 < \omega < 1 \)
In this kind of face pair, the type of matching face is the same as the corresponding face in source feature, but there are differences in the aspect of size or manufacturability.
Algorithm 3 Face Adjacency Judgment Procedure

1. **Input:** Optimal matching M
2. **Output:** judgment if there is isolated face; if yes, the first isolated face is found
3. Face set T ← faces retrieved from M, removing faces \( \omega = 0 \)
4. \( m \leftarrow \) face number in set T
5. for \( (i = 1; i \leq m; i++) \) do
6. Face set A ← faces in set T except No. \( i \) face
7. Contour set C ← Retrieve contours in set A
8. Edge set \( E_i \leftarrow \) Edges in set C
9. Edge set \( E_j \leftarrow \) Edges of No. \( i \) face
10. \( n \leftarrow \) Edge number of No. \( i \) face
11. \( j \leftarrow 1 \)
12. for \( (j \leq n) \) do
13. \( e_j \leftarrow \) No. \( j \) edge of set \( E_j \)
14. if \( (e_j \in E_i) \) then
15. break
16. end if
17. end for
18. if \( (j == n+1) \) then
19. return No. \( i \) face is isolated
20. end if
21. end for
22. return there is no isolated face in set T

3) THE MATCHING WEIGHT IS 0

In this condition, the type of matching face is not the same as its corresponding face in source feature. According to the Rule 1, its weight is 0.

Based on the preceding three kinds of face matching, the average value of the weight between every matching face pair is used as the similarity between source feature and the local feature of target model, and the formula is as follows,

\[
Sim_{S,T} = \frac{1}{m} \times \sum_{i=1}^{m} (\omega_{S_i,T_i})
\]  

(29)

where, \( Sim_{S,T} \) is the similarity evaluation value between source feature and the local feature of target model. \( \omega_{S_i,T_i} \) is the weight between the No. \( i \) pair of matching faces. \( m \) is the number of matching face pairs between the two models after excluding supplementary faces.

VI. CASE STUDY

To validate the proposed feature recognition and similarity evaluation algorithm, case studies are performed in collaboration with a machinery enterprise. This enterprise is specialized in mechanical product design and manufacturing. After years of accumulation, a database of product models and production processes has been formed. Due to the requirements of design, processing and assembly, it is often necessary to retrieve local features similar to the current design, such
as the appearance and manufacturability of models in the existing model, so as to find similar production processes. It is necessary to find a model retrieval method, which is utilized to traverse the existing model in the database to search the local features similar to the selected features in current source model. By evaluating the similarity, we can find the model which reaches the specific similarity threshold. Based on the preceding application background, the recognition and evaluation of features are taken as example to verify the proposed methods in this paper.

A. CASE 1

In this case, the model shown in Fig.7 is taken as source model, and the stepped shaft feature, shown in the circle of source model, is taken as source feature. The model shown in Fig.8 is taken as target model, and the local feature similar to the source feature are searched from the target model and their similarity is evaluated.

Select source feature in Creo, and then start feature recognition and similarity evaluation program. The basic information of source feature and target model is identified, as shown in Fig.9-12. MDAAM of the source feature are established as shown in supporting materials as Formulas (S1) - (S6), and MDAAM of the target model are shown in supporting materials as Formulas (S7) - (S12).

In this example, the source feature has 16 faces and the target model has 31 faces. So, 15 supplementary faces are added to the source feature. Weighted complete bipartite graph is constructed by using the faces of source feature and target model as independent sets and the weight matrix of the graph is obtained, shown in Formula (30), as shown at the bottom of the page.

Kuhn-Munkres algorithm is used to retrieve the optimal matching in the bipartite graph. After removing supplementary faces, the matching relationship between the faces of source feature and target model is shown as shown in Formula (31).
Algorithm 4: Face Adjacency Judgment and Isolated Face Processing Algorithm

1. Input: weighted complete bipartite graph \( G = (V_S, V_T, E) \) and its weight matrix \( WM \) retrieved face set \( T (\omega \neq 0) \) from perfect matching \( M \) of target model isolated face \( e \) in set \( T \) (retrieved in Algorithm 3)

2. Output: perfect matching \( M \) without isolated face

3. \( m \leftarrow \) Face number of source feature

4. while (1)

5. for (i = 0; i \leq m; i++)

6. \( \omega'(S(i), TS(isosurf)) \leftarrow \omega(S(i), T(isosurf)) \times (1 - \alpha) \)

7. end for

8. call Algorithm 2

9. remove supplementary faces

10. call Algorithm 3

11. if (isolated face \( e \) found) then

12. continue

13. else

14. break

15. end if

16. end while

17. return

in Fig. 13. At the same time, the figure shows the source feature and its matching local feature found in the target model.

According to the adjacency judgment, the face with ID 536 in target model is an isolated face, which corresponds to No.27 column of the weight matrix. Therefore, the weights of the No.27 column matrix elements shown in Formula (30) are reduced to obtain a new weight matrix as shown in Formula (31), as shown at the bottom of the previous page.

The reduced weight matrix, shown in Formula (31), is used to perform the optimal matching again. After removing the supplementary faces, the corresponding relationship between the faces of source feature and target model in optimal matching is shown in Fig.14. Through adjacency judgment, there is no isolated face in the target model faces set, which meets the requirements of feature recognition.

According to Formula (29), the similarity between the source feature and the local feature retrieved from target model is 0.81, and the feature recognition and similarity evaluation are finished.

B. CASE 2

In this case, a complex product with multiple heavily intersecting features is used to test the proposed method. The highlighted surfaces in Fig.15 are taken as source feature, and the model shown in Fig.16 is taken as target model.

The basic model topological information is extracted as shown in Fig.17-Fig.20, and the database is established.
according to the information of source features and target model faces and edges, as shown in Fig.21-Fig.24. According to the proposed method of feature recognition, MDAAM and weight matrix of weighted complete bipartite graph is constructed, and the optimal matching of bipartite graph is calculated by KM algorithm.

After removing the supplementary faces, the corresponding relationship between the faces of source feature and target model in optimal matching is shown in Fig.25. According to the adjacency judgment, the face with ID 87 in target model is an isolated face. Reduce its weight and match again. The corresponding relationship is shown in Fig.26 and there is no isolated face in the target model faces set, which meets the requirements of feature recognition. The similarity between the source feature and the local feature retrieved from target model is 0.970.

C. CASE 3

In order to verify the performance of the proposed algorithm, it is compared with the method proposed in [13] to evaluate the similarity of the model. Using the workstation Intel (R) Xeon (R) CPU E5-2630 V3 @ 2.40GHz as the
experimental platform, 200 Creo models are selected as test model library, one of which is selected as the source model, and the similarity between the other models in the model library and the source model is calculated by the similarity evaluation method mentioned above.

In terms of time-consuming, the proposed algorithm is divided into two parts: model information preprocessing time-consuming and similarity computing time-consuming. Among them, the preprocessing of model information includes reading model data, establishing information database, and the building of MDAAM, which takes a large amount of time.

In this paper, we first preprocess the model, and then reuse it. Due to the lack of information storage, the method described in [13] needs to re extract the model information every time the model is used.

Table 5 shows the comparison of the time consumption of different algorithms. It can be seen that the time consumption of the proposed algorithm is larger when the model information is not preprocessed, but the time consumption is significantly reduced after the model preprocessing is completed.

**TABLE 5. Comparison of time consumption.**

| Method                                      | Time consumption(s) |
|---------------------------------------------|---------------------|
| Proposed algorithm without information preprocessing | 262                 |
| Information preprocessing of proposed algorithm     | 125                 |
| Similarity calculation of proposed algorithm      | 137                 |
| Similarity calculation method in [13]            | 182                 |

**VII. CONCLUSION**

In summary, a novel method of feature identification and similarity evaluation based on MDAAM is proposed. On the basis of extracting B-rep model information, MBD based MDAAM is constructed, which contains source feature and target model. A weighted complete bipartite graph is constructed by taking the face sets of source feature and target model as independent sets and the similarity between faces as weights. Kuhn-Munkres algorithm is used to calculate the optimal matching between the two independent sets to identify the matching degree between local features and source features in target model. Finally, the adjacency of the matching faces is judged and the similarity of matching faces is evaluated. Taking stepped shaft parts as example, the recognition of local feature in target model is realized. Then the similarity between local feature and source feature is calculated accurately from the aspects of topology, shape and size, manufacturability, etc., which verifies the effectiveness of the proposed method.

Position tolerance is a kind of multi-element annotation, which cannot be read and used directly through the information of a face or an edge. In the future, we will focus on the position tolerance between surfaces. Based on this, the similarity of process information and its evaluation method will be further studied.

**REFERENCES**

[1] M. Al-Wswasi, A. Ivanov, and H. Makatsoris, “A survey on smart automated computer-aided process planning (ACAPP) techniques,” Int. J. Adv. Manuf. Technol., vol. 97, nos. 1–4, pp. 809–832, Jul. 2018, doi: 10.1007/s00170-018-1966-1.

[2] B. S. Prabhu, “Automatic extraction of manufacturable features from CAD models using syntactic pattern recognition techniques,” Int. J. Prod. Res., vol. 37, no. 6, pp. 1259–1281, Apr. 1999, doi: 10.1080/002075999191247.

[3] B. R. Babic, N. Nešić, and Z. Miličković, “Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems,” Artif. Intell. Eng., Anal. Manuf., vol. 25, no. 3, pp. 289–304, Aug. 2011, doi: 10.1017/S0900970010000545.

[4] A. Cardone, S. K. Gupta, and M. Karmic, “A survey of shape similarity assessment algorithms for product design and manufacturing applications,” J. Comput. Inf. Sci. Eng., vol. 3, no. 2, pp. 109–118, Jun. 2003, doi: 10.1115/1.1577356.

[5] K. X. Zhang, S. S. Zhang, and X. L. Liu, “Current research and future development of 3-D CAD model retrieval,” Trans. Chin. Soc. Agric., vol. 44, no. 7, pp. 256–263, Mar. 2013.

[6] S. Joshi and T. C. Chang, “Graph-based heuristics for recognition of machined features from a 3D solid model,” Comput.-Aided Des., vol. 20, no. 1, pp. 59–66, Mar. 1988.

[7] Z. Gu, Y. F. Zhang, and A. Y. C. Nee, “Generic form feature recognition and operation selection using connectionist modelling,” J. Intell. Manuf., vol. 6, no. 4, pp. 263–273, Aug. 1995.

[8] L. Liu, Z. Huang, W. Liu, and W. Wu, “Extracting the turning volume and features for a mill/tum part with multiple extreme faces,” Int. J. Adv. Manuf. Technol., vol. 94, nos. 1–4, pp. 257–280, Jan. 2018, doi: 10.1007/s00170-017-0862-4.

[9] M. El-Mehalawi and R. A. Miller, “A database system of mechanical components based on geometric and topological similarity—Part I: Representation,” Comput. Aided Des., vol. 35, pp. 83–94, 2003, doi: 10.1016/S0010-4485(01)00177-4.

[10] M. El-Mehalawi, “A database system of mechanical components based on geometric and topological similarity—Part II: Indexing, retrieving, matching, and similarity assessment,” Comput. Aided Des., vol. 35, pp. 95–105, 2003, doi: 10.1016/S0010-4485(01)00178-6.

[11] K. X. Zhang, S. Y. Hang, J. X. Wang, Z. H. Song, and X. L. Liu, “Typical structure mining in 3D model and similarity evaluation based on simulated annealing algorithm,” Trans. Chin. Soc. Agric., vol. 49, no. 3, pp. 402–410, Mar. 2018.

[12] X. M. Liu, Y. Q. Jia, Z. R. Chen, and A. P. Li, “Recognition of machining feature for engine cylinder blocks,” Comput. Integr. Manuf. Syst., vol. 22, no. 5, pp. 1197–1204, 2016.

[13] H. S. Wang, S. S. Zhang, X. L. Bai, and Y. Z. Zhang, “Optimal-matching-based 3D CAD model similarity assessment algorithm,” Comput. Integr. Manuf. Syst., vol. 13, no. 10, pp. 1921–1927, 2007.

[14] J. Oussama, E. Abdelilah, and R. Ahmed, “Manufacturing computer aided process planning for rotational parts—Part 1: Automatic feature recognition from STEP AP203,” Int. J. Eng. Res. Appl., vol. 4, no. 5, pp. 14–25, 2014.

[15] L. Zehtabian and D. Roller, “Automated rule-based system for opitz feature recognition and code generation from STL,” Comput.-Aided Des. Appl., vol. 13, no. 3, pp. 309–319, 2016, doi: 10.1016/j.cadap.2015.11.14388.

[16] G. Campana and M. Mele, “An application to stereolithography of a feature recognition algorithm for manufacturability evaluation,” J. Intell. Manuf., vol. 31, no. 1, pp. 199–214, Jan. 2020, doi: 10.1007/s10845-018-1441-8.

[17] S. M. Gao, “A survey of automatic feature recognition,” Chin. J. Comput., vol. 21, no. 3, pp. 281–288, 1998.

[18] L. Luo, M. Zhou, Y. Lu, T. Yang, and F. Yang, “A hybrid 3D feature recognition method based on rule and graph,” Int. J. Comput. Integr. Manuf., vol. 34, no. 3, pp. 257–281, Mar. 2021, doi: 10.1080/0951192X.2020.1858507.

[19] V. S. Sunil, R. Agarwal, and S. S. Pande, “An approach to recognize interacting features from B-rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique,” Comput. Ind., vol. 61, no. 7, pp. 686–701, Sep. 2010, doi: 10.1016/j.compind.2010.03.011.

[20] Y. Li, W. Wang, X. Liu, and Y. Ma, “Definition and recognition of rib features in aircraft structural part,” Int. J. Comput. Integr. Manuf., vol. 27, no. 1, pp. 1–19, Jan. 2014, doi: 10.1080/0951192X.2013.799784.
[21] S. Gao and J. J. Shah, “Automatic recognition of interacting machining features based on minimal condition subgraph,” Comput.-Aided Des., vol. 30, no. 9, pp. 727–739, Aug. 1998.

[22] V. Rameshbabu and M. S. Shunmugam, “Hybrid feature recognition method for setup planning from STEP AP-203,” Robot. Comput.-Integr. Manuf., vol. 25, no. 2, pp. 393–408, Apr. 2009, doi: 10.1016/j.rcim.2007.09.014.

[23] A.-J. Fougères and E. Ostrosi, “Intelligent agents for feature modellling in computer aided design,” J. Des. Eng., vol. 5, no. 1, pp. 19–40, Jan. 2018, doi: 10.1016/j.jdec.2017.11.001.

[24] B. R. Borkar and Y. M. Puri, “Automatic extraction of machining features from prismatic parts using STEP for downstream applications,” J. Inst. Eng. (India), Ser. C, vol. 96, no. 3, pp. 231–243, Jul. 2015, doi: 10.1007/s40032-015-0171-3.

[25] W. D. Li, S. K. Ong, and A. Y. C. See, “A hybrid method for recognizing interacting machining features,” Int. J. Prod. Res., vol. 41, no. 9, pp. 1887–1908, Jan. 2003, doi: 10.1080/0020754031000123868.

[26] Z. Zhang, P. Jaiswal, and R. Rai, “FeatureNet: Machining feature recognition and measurement path generation system based on NURBS surface descriptions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 11, pp. 1140–1157, Nov. 1989.

[27] X. Y. Gao, H. N. Li, C. X. Zhang, and X. Y. Yu, “Similarity calculation technique,” Comput.-Aided Des., vol. 104, nos. 1–4, pp. 261–284, Sep. 2019.

[28] Y. Shi, Y. Zhang, S. Baek, W. De Backer, and R. Harik, “A novel learning-based feature recognition method using multiple sectional view representation,” J. Intell. Manuf., vol. 31, no. 5, pp. 1291–1309, Jun. 2020, doi: 10.1016/j.intmat.2018.02.151.

[29] X. Y. Gao, H. N. Li, C. X. Zhang, and X. Y. Yu, “Similarity calculation of 3D CAD model based on ant colony searching,” J. Southwest Jiaotong Univ., vol. 52, no. 2, pp. 416–423, 2017.

[30] J. Wang, “Research on shape feature recognition of B-rep model based on wavelet transform,” Math. Problems Eng., vol. 2018, pp. 1–8, Jul. 2018, doi: 10.1155/2018/610482.

[31] B. C. Kim and D. Mun, “Enhanced volume decomposition minimizing overlapping volumes for the recognition of design features,” J. Mech. Sci. Technol., vol. 29, no. 12, pp. 5289–5298, Dec. 2015, doi: 10.1007/s12206-015-1311-9.

[32] D. Bespalov, W. C. Regli, and A. Shokoufandeh, “Local feature extraction and matching partial objects,” Comput.-Aided Des., vol. 38, no. 9, pp. 1020–1037, Sep. 2006.

[33] Y. Shi, Y. Zhang, S. Baek, W. De Backer, and R. Harik, “Manufacturability analysis for additive manufacturing using a novel feature recognition technique,” Comput.-Aided Des. Appl., vol. 15, no. 6, pp. 941–952, Nov. 2018, doi: 10.1016/j.cada.2018.04.057.

[34] M. Al-wswassi and A. Ivanov, “A novel and smart interactive feature recognition system for rotational parts using a STEP file,” Int. J. Adv. Manuf. Technol., vol. 104, nos. 1–4, pp. 261–284, Sep. 2019.

[35] Y. Yoo and S. H. Kim, “Protrusion recognition from solid model using globally G1triangular splines,” Comput. Aided. Geom. Des., vol. 77, pp. 1–16, Feb. 2020, doi: 10.1016/j.cagd.2020.101832.

[36] D. B. Pereng, Z. Chen, and R. K. Li, “Automatic 3D machining feature extraction from 3D CSG solid input,” Comput. Aided. Des., vol. 22, no. 5, pp. 285–295, 1990, doi: 10.1016/0010-4485(90)90093-R.

[37] B. Venu, V. R. Komma, and D. Srivastava, “STEP-based feature recognition system for B-spline surface features,” Int. J. Automot. Comput., vol. 15, no. 4, pp. 500–512, Aug. 2018, doi: 10.1016/j.jmac.2016.11.001.

[38] S. Nagarajan and N. V. Reddy, “STEP-based automatic system for recognising design and manufacturing features,” Int. J. Prod. Res., vol. 48, no. 1, pp. 117–144, Jan. 2010, doi: 10.1080/00207540701855419.

[39] A. S. M. Hoque, P. K. Halder, M. S. Parvez, and T. Szcesi, “Integrated manufacturing features and design-for-manufacture guidelines for reducing product cost under CAD/CAM environment,” Comput. Ind. Eng., vol. 66, no. 4, pp. 988–1003, Dec. 2013, doi: 10.1016/j.cie.2013.08.016.

[40] B. Venu and V. R. Komma, “STEP-based feature recognition from solid models having non-planar surfaces,” Int. J. Comput. Integr. Manuf., vol. 30, no. 10, pp. 1011–1028, Oct. 2017, doi: 10.1080/0951197X.2016.1268719.

[41] T. J. Fan, G. Medioni, and R. Nevatia, “Recognizing 3-D objects using surface descriptions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 11, pp. 1140–1157, Nov. 1989.

[42] C. H. Li, Z. J. Li, and Y. Gao, “Application of spectrum and Kuhn–Munkres algorithm in graph matching,” Comput. Eng. Sci., vol. 39, no. 10, pp. 1896–1900, 2017.

SHUHUI DING was born in Shandong, China, in 1977. He received the Ph.D. degree in mechanical engineering from the Shandong University of Science and Technology in 2017. Since 2012, he has been working as an Associate Professor with the Mechanical Manufacturing Department, Shandong University of Science and Technology. Since 2020, he has also been a Visiting Scholar with the Department of Mechanical Engineering, University of California at Berkeley. He is the author of eight books, five inventions, and more than ten articles. His research interests include digital design, mechanical CAD, cloud manufacturing, intelligent manufacturing, and industrial robot integrated systems.

QIANG FENG was born in Shandong, China, in 1996. He is currently pursuing the master's degree in mechanical engineering with the College of Mechanical and Electronic Engineering, Shandong University of Science and Technology, China. His research interests include digital design, computer graphics, and model feature recognition.

ZHAOYANG SUN received the B.Eng. and M.Eng. degrees from the Shandong University of Science and Technology, Qingdao, China, and the Ph.D. degree from Beihang University, Beijing, China. Since September 2009, she has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Research Fellow. Since July 2013, she has also been with the China National Institute of Standardization, Beijing, as an Associate Professor. Her main research interests include design rationale, data mining, and e-commerce.

FAI MA received the B.S. degree from The University of Hong Kong, in 1977, and the Ph.D. degree from the California Institute of Technology, in 1981. He was a Research Engineer with the IBM Thomas J. Watson Research Center and Standard Oil Company. He is currently a Professor of applied mechanics with the Department of Mechanical Engineering, University of California at Berkeley. For over 30 years, he has been engaged in research in the areas of vibration, nonlinear damping, and system uncertainties. He is the author or coauthor of more than 190 technical publications and is a fellow of the American Society of Mechanical Engineers.