www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Critical Review of Feature Recognition Techniques

Yang Shi[1], Yicha Zhang[2], Kaishu Xia[3] and Ramy Harik[4]

[1]University of South Carolina, yshi@email.sc.edu
[2]Université de Technologie de Belfort-Montbéliard, Yicha.Zhang@ utbm.fr
[3]University of South Carolina, kxia@email.sc.edu
[4]University of South Carolina, harik@cec.sc.edu

Corresponding author: Ramy Harik, harik@cec.sc.edu

Abstract. Feature is an essential concept during design and manufacturing. It has universal characters for specific definitions of interest. Feature Recognition (FR) is a technique to identify and extract application-specific information from input models for downstream engineering activities. FR is a necessary and important component for the integration of Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM), Computer-Aided Engineering (CAE), and Computer-Aided Process Planning (CAPP). Decades of research work in this topic has developed numerous techniques such as rule-based, graph-based, volume decomposition, and artificial neural network-based systems. However, there still exist issues that hinder FR to become a practical engineering tool. This paper presents a critical literature review of these approaches.

Keywords: Feature Recognition, Feature Extraction, CAD/CAM
DOI: https://doi.org/10.14733/cadaps.2020.861-899

1 INTRODUCTION

Feature is one of the most common and indispensable notions in computer-aided design (CAD) and other engineering activities in industry. It is a representative tool to capture expert knowledge accomplishments, and to enable an advanced information flow from early design to all phases within product lifecycle [78]. However, mainstream CAD systems mainly focus on creating geometric shapes by Boolean operations with a set of CAD features, and the notion of manufacturing characteristics are not involved. For example, a cylinder protrusion is a common design feature in CAD, but it cannot be manufactured by a single machining operation, for which reason cannot be identified as a single machining feature. Feature specifies designed parts at a higher level of description than engineering drawings or CAD models. It characterizes the engineering significances in terms of mathematical forms, which facilitate the automation of engineering activities. In addition, it can also provide a high-level interface for the CAD systems. Using the notion of feature, designer can modify the shape configuration in CAD models by setting corresponding parameter values. Feature also can be deemed as a communication medium between design and manufacturing. With the use of features, parameterized shape information from CAD models can directly support the manufacturing planning in a CAM system [65].

Feature recognition is a critical sub-discipline of CAD/CAM that focuses on the design and implementation of algorithms for detecting manufacturing significance from CAD models [38]. It can be considered as a necessary and fundamental component to automate and integrate design and downstream applications such as engineering analysis [36] [105], optimization [2] [144], design validation [37], manufacturing planning [74] [79] [88] [131] or DFX consideration [28] [89], etc. Economically speaking, feature recognition techniques can facilitate the various engineering tasks to dramatically reduce the total life cycle of the product development [84] [89]. Moreover, it can meet the needs of different abstractions of the same product for different development levels and activities during product lifecycle [78].

In particular, feature recognition technology plays an indispensable role in production engineering. Feature is one of the most common and important notions for technical communication and decision-making in the industrial activities. Recognition means awareness of these known expert knowledge and refining for downstream activities. Feature recognition techniques are required for manufacturing since the CAD models cannot be used directly by these engineering analysis and decision-making systems because CAD models lack high-level geometrical or topological entities that are meaningful for these applications [20]. Therefore, the key actions occurring in feature recognition is translating the low-level geometric entities from the CAD models into a set of appropriate 'features' with inherent attributes related to design intent and manufacturing functions [49] [134]. Although many different definitions of features have been given, the nature of features can be identified as a higher-level description of the design than traditional drawings or CAD models, which characterizes the engineering significance of designs in term of mathematical surfaces or volumes.

At a first glance, recognizing features looks like an intuitive task since the geometric features in CAD models seem apparent to human eyes. Because analyzing spatial object visually is a born natural skill for humans, with a superior brain capability in spatial reasoning [50]. Nonetheless, the automatic recognition by computer is a tough problem. Specifically, features are difficult to be characterized than to be recognized. Human can identify features even if the feature characterization and classification remain intuitive and fuzzy, but computers need a definite and mathematical characterization can be carried out by algorithms [48]. In consideration of the shape variations and changeable dimensions, single feature definition could be already inexhaustible, not to mention the definition of feature interactions.

There is also another way to realize the translation between design intent and manufacturing features, which is the design-by-feature (DBF) approach. In this method, the product models are formed by selecting features from a predefine feature library, which is accommodated not to the functional meaning, but to the manufacturing constraints. However, a few disadvantages make it less competitive than feature recognition. For example, it limits the designer's creativity and scope by requiring them to have a clear awareness of the production environment [20]; and it is difficult to include all manufacturing features in reality. In addition, even during the design process with DBF, due to feature interactions, feature recognition and validation techniques are still needed to check whether the newly added features would affect the validity of existing features [67] [98].

During last three decades, various feature recognition systems have been developed such as logic rules, graph-based, volume decomposition, hint-based, neural network, and hybrid approaches. As shown in Figure 1, the research activities of feature recognition techniques have passed the peak and been in the period of bottom. According to the theory of Gartner hype cycle, the technology of feature recognition is about getting into the "slope of enlightenment" that it will start to benefit enterprise. Nevertheless, there are still drawbacks that hinder its practical applications, such as lack of robustness, inability to learn, limited domain of features, and computational complexity. In spite of these problems, the interest in feature recognition research has decreased over past few years, probably because the need of platform-independent CAX tools for arbitrary shapes is reducing. The main goal of this paper is to conduct a literature review of these recognition methods with their strengths and weaknesses. In further perspective of this research, an original method developed for feature recognition functioning with increased efficiency is planned.

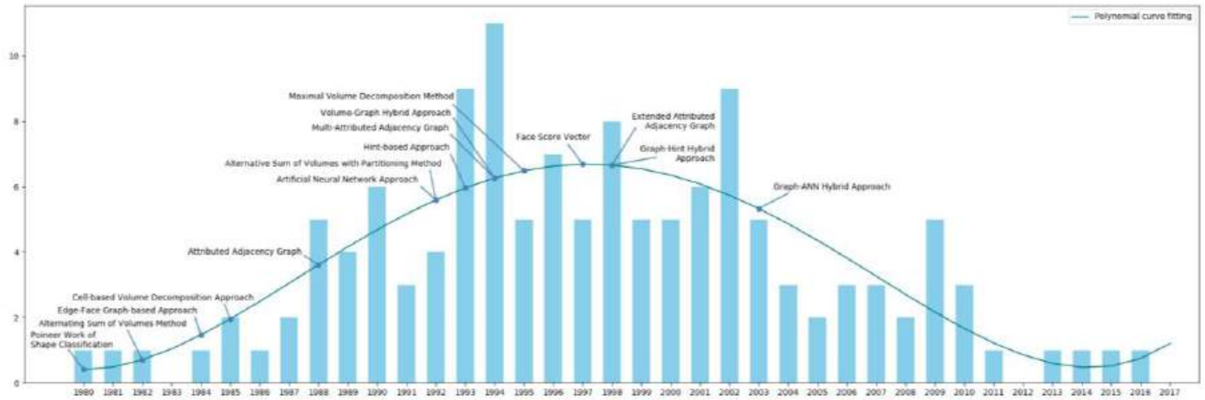


Figure 1: The number of reviewed literatures in each year and milestones of feature recognition technique.

In summary, most of the feature recognition systems have three basic modules [140]: (1) Feature definition, (2) Feature representation, and (3) Feature recognition mechanism. The feature definition module is intended to decompose the feature space into several expected classes. Each class denotes some specific needs for the downstream engineering tasks. The conditions for defining a class are generally described by a set of elements in a representation scheme, usually by sampling some typical examples and characterizing them. In the feature representation module, the low-level geometric information in input CAD model is converted into high-level form with geometric and topological attributes to facilitate the feature searching/matching. In the feature recognition mechanisms, features are first extracted by certain separation rules for further analysis. For example, in a graph-based approach, the graph of part can be divided into subgraphs based on edge convexity. Then typical pattern recognition techniques would be developed correspondingly to be compatible with the feature representation scheme. Finally, the recognition phase identifies the semantics of the feature. In the following text, these three modules are discussed in sequence.

2 DEFINITION OF FEATURES

Feature is a necessary engineering tool for geometric and semantic reasoning about design shapes. Because of the wide use, it is also one of the most diversely defined notions. There has not been a universal and consensus definition can be found among the academic or research community [5]. Basically, these definitions are based on the specific needs and interest of the researchers. Generally, they can be classified into two main categories: implicit and explicit.

2.1 Implicit Definition

Implicit definition of feature, which is also known as form feature in most cases, gives a conceptual feeling about a certain pattern of the part. It could be comprehensive and inclusive but cannot be directly implemented by algorithms. From a boarder perspective, [133] named such "a region of interest on the surface of a part" as form feature, which is useful in various applications. [106] detailed this concept as "a single face or a set of connected face with certain characteristic combinations of topology and geometry". [35] defined feature as "local geometric entities" and form feature is “a shape pattern has some significance” and claimed how to represent feature is greatly dependent on the application. Thereafter, many similar definitions are proposed in the research of feature recognition, and they all involved with “shape”, “characteristics”, “significances” and “applications”, such as [20],[28],[89],[111],[112],[121],[124].

2.2 Explicit Definition

In order to apply to various downstream engineering activities, feature implicit definitions have to be decomposed into certain subclasses depending upon their intended applications, such as design features, manufacturing features, assembly features, and analysis features [96]. This classification is also a further clarification about the domain characteristics that investigated researches want to recognize. Therefore, definitions of various types of explicit features are proposed. For example, from the viewpoint of shape modeling, features can be defined as a collection of modeling primitives, which are parameterized by topological and/or geometric variables [35],[82],[95]. While considering applicable to feature recognition, features are defined based on the characteristic patterns in topology and geometry, such as orthogonal/non-orthogonal, polyhedral/cylindrical, interior/exterior, convex/concave, surface/volume feature and so on [28],[64],[88],[134]. The following figure 2 is an example of sub-classification of form features.

In existing feature recognition systems, the definition normally includes a shape representation and a parametric representation. Therefore, the feature would be a precise description on a specific shape so that it can be represented in a computational form and recognized algorithmically on a computer [140]. To be robust and to avoid ambiguity, explicit definition should include the minimal set of necessary conditions that classify a feature uniquely [51]. These conditions are extracted from the particular characteristics of topological and geometric information and form a representation scheme of the feature boundary. For instance, [35] describes the features using five geometric conditions: protrusion/depression, the number of tool accessible directions from which a feature can be machined, through or not, boundary perimeter and boundary geometry. Nine different features are given in the literature. Moreover, a secondary class can be defined by indicating the perimeter geometry, such as square pocket, cylinder hole, and dovetail slot and so on. The methods for proposing feature representation scheme will be discussed in the next section.

2.3 Manufacturing Features

Manufacturing feature is the most common objective of feature recognition. It represents a set of solid geometric spaces in the part that are generally manufactured by shape-forming or material-removing processes, such as machining feature, forging feature, casting feature and so on. They are widely used for manufacturability analysis and process planning activities such as the configuration of work piece holding, choice of machines and cutting tools, and planning of the machining operations [70].

Among manufacturing features, the most discussed one is machining feature, though there are several implementations of feature recognition techniques in other manufacturing domains, such as sheet metal stamping [12] [69] [116], molding [76] [81], forging and casting [53]. At the early stage, the definition was only a simple description of the relationship between feature and machining process. For example, it was defined as a sequence of connected surfaces generated by material removing processes in [13]. Along with the development of feature recognition technique, the definitions became more specific that it was seen as the corresponding volume that removed by a single or a sequence of operations, because of the demand for application to process planning [140].

In addition, [88] defines the solid geometric space as primary feature, and the removed geometric space around or in the primary feature as secondary feature. [28] defined the stock volume as a bounding box: “the minimal volume of a cubic block containing the designed part”. Then the volume in the definition of machining feature becomes the “difference between the bounding box and the designed part”.

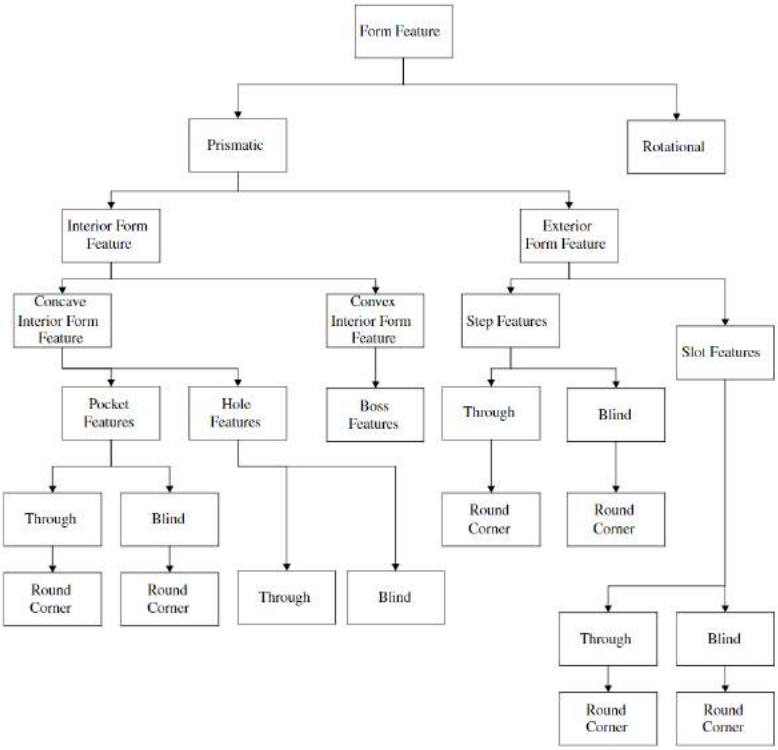


Figure 2: Hierarchy of Form Feature [89].

[138] gave a more rigorous definition of machining feature, which includes three conditions:

- It is contained in the delta volume, which refers to the volumetric difference between the raw stock and the final part;
- It can be removed from the workpiece by one machining operation with a 3-axis machining center. One machining operation is defined as a movement of one cutter in one setup without retracting from the workpiece;
- Its removal creates a portion of the part's surface without destroying the part.

STEP AP 224 [45] is an ISO STEP application protocol to specify the manufacturing information for process planning in terms of manufacturing features. In the standard, manufacturing features are defined as shape representations that describe volumes of materials that shall be removed from a part by machining or shall result from machining. From the manufacturing viewpoint, AP 224 provides a universal feature library with technological attributes attached, and a systematic feature classification scheme including four types of feature: machining features, transition feature, replicate feature and compound feature. Figure 3 shows some examples of the 16 machining features defined in AP 224.

2.4 Transition Features

Transition features are special surface features generated by the trimming and blending edges or releasing vertices in the part boundary. The operations to trim and blend edges or release vertex convert some edges and vertices within a part into corresponding surfaces, and these surfaces or their combinations constitute the transition features [28]. These features, namely fillet, chamfer and round (Figure 4), are essentially existing in everywhere of the real-world mechanical parts as the guarantee of design manufacturability and assemblability [148].

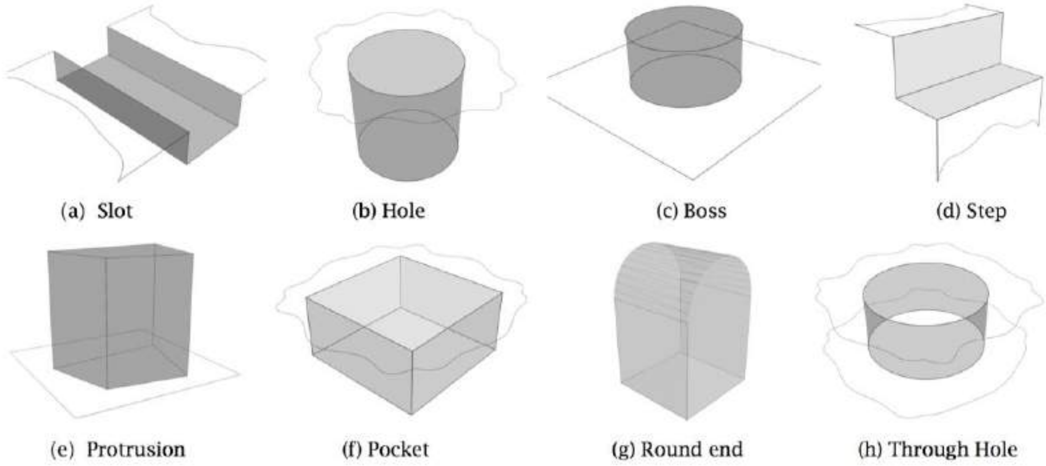


Figure 3: Eight examples of machining feature in STEP AP 224.

With a large amount of small transition features, the shape of CAD models becomes complicated as the geometric and topological representation of part changes. Similar to interacting features, because of boundary variations, transition features arise troubles during the extraction and decomposition from complex features to primitive volumetric features. Moreover, when transition features interact with volume features, although the general shape of primary features may not be affected, the results of feature recognition may become even worse.

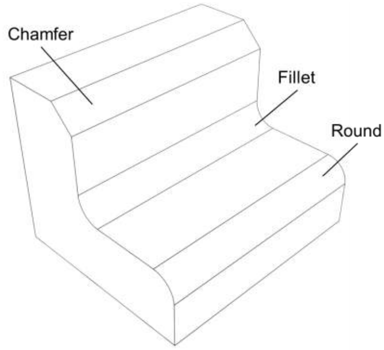


Figure 4: Transition Features.

In CAD modeling, transition features are also referred as secondary features because they are only used to modify the boundary conditions of primary features such as holes, slots and pockets. In fact, they are not the key intentions for either design or manufacturing. Thus, it is desirable to suppress these features without any effect to the primary features to facilitate feature recognition. In addition, the fillets and rounds produce many small no-linear surfaces such as cylindrical, spherical and toroidal surfaces, which are unnecessary for finite element analysis (FEA), but prolong FEA mesh generation time, and impair FEA mesh quality. Therefore, in order to improve the efficiency and accuracy of FEA, CAD mesh model simplification should suppress transition detailed features without any changes to the rest [31]. The transition features in machining parts are not difficult to be extracted, because under some circumstances one can assume that only them having curved surfaces in the part. A representative approach was proposed by [148] to simplify B-Rep models by automatic fillet/round suppressing. The fillet/round features are identified based on topological characteristics, then fillets and rounds are replaced by edges using incremental knitting method. Nevertheless, only constant-radius fillet and rounds are tested in this work.

2.5 Feature Interaction

If the boundaries of different volumetric features intersect, they together can be defined as interacting features [30]. It is acknowledged as the most critical issue in the field of feature recognition. The ability to handle interacting features has been an informal benchmark for a feature recognition system [127]. On one hand, feature interaction is a major challenge in the development of feature recognition approaches with robustness [71], especially for popular feature recognition methods such as graph-based and hint-based feature recognition, which rely on tracing the typical topological characteristics. Because difficulties of recognizing interacting features arise from the fact that partial or all of the boundaries are divided into several segments or even destroyed, and the characteristics of pre-defined features is missing [88].

On the other hand, interacting features have to be further broken down to primitive features so that they can be used by downstream engineering activities. For most feature recognition techniques, the feature definitions are explicit and unique. However, a complex interacting feature could have multiple interpretations, in other words, the feature can be decomposed in different ways. In that case, it is even a tougher task for human intuition. The ambiguity caused by multiple interpretations has not been perfectly resolved. Figure 5 describes an example of multiple interpretations of volumetric machining features.

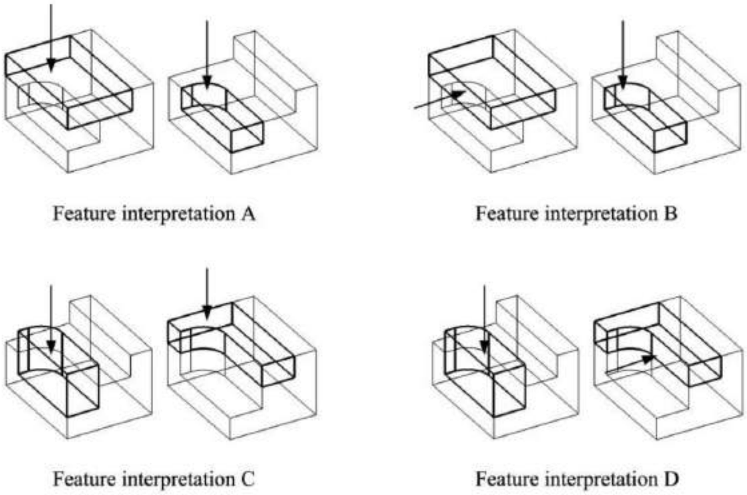


Figure 5: Multiple sets of feature interpretation [138].

The most adopted definition of the various types of volumetric feature interactions is provided by [30]. As shown in Figure 6, from the topological point of view, six types of interactions only involve three classes of topological variations: merging of faces, loss of concave edges, and splitting of faces. Nevertheless, in [65], the interactions are viewed based on faces. Three different types of interaction are taken into definition: 1. One feature sits on top of another feature, so all or part of a face for both features is missing. 2. Both features have the same base surface and their areas overlap. In this case, the part of the base surface where the areas intersect is influenced by parameters from both features. 3. Both features have the same base surface, but their areas do not overlap. In this case, a combination of the two features results in a pattern feature.

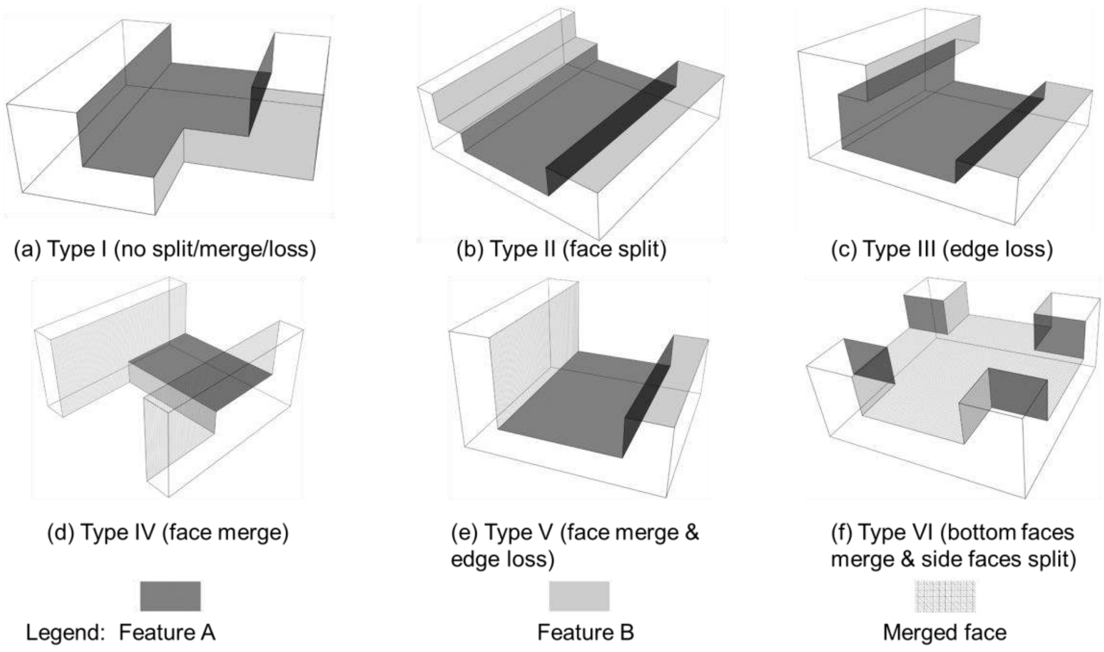


Figure 6: Examples of different types of feature interactions [30].

However, for these existing solutions of feature interaction, only the cases of two orthogonal features interacting are considered. When more features interact together, it is not possible to define the interacting patterns for each and every conceivable feature interaction. Not to mention the non-orthogonal and freeform features, they can interact in infinite ways and create irregular volumes, which cannot be recovered using the remaining geometry.

3 FEATURE REPRESENTATION

Traditionally, engineers use engineering drawings to convey information about the part's shape, geometry, and other key attributes necessary for successful manufacturing of the part [50]. The current generation of CAD systems places emphasis on creating complex surfaces and solids by Boolean operations with a library of primitive shapes to increase the productivity of the designer. However, the information stored in the model is at a primitive level in the form of basic geometric entities and topology [124]. It does not represent the notion of features explicitly, hence is not applicable to the downstream engineering activities.

Feature representation is an essential component for feature recognition system. It is a translation from the low-level geometric entities to interested high-level description of functional shapes. Feature recognition is the process of identifying expected patterns of geometric entities, corresponding to particular engineering significances in part models [2]. Therefore, the basic task of feature recognition system is searching common high-level information, which can be used to classify the geometric patterns/features, from the set of lower-level entities of a part model [103].

Most of existing feature recognition approaches recognize features by matching the entities composing a feature, together with their interrelationships, against certain predefined set of rules or templates. The idea for building such a set (library) of feature templates is following a 'sampling-and-characterization' process. That is, typical feature classes are first sampled as templates and their common characteristics are summarized subsequently in a designed feature representation scheme.

The common high-level information is the characteristic attributes of topological, geometrical and hierarchical information about the part. The topological information describes the connectivity and associativity between the entities. There are 25 kinds of adjacency relationships, each one has a unique ordered pair of topological entities [132]. The geometrical information specifies the dimension and location of each topological element. Using the chosen characteristic attributes, we can describe, classify and map the features in a representation scheme.

The topological, geometrical and hierarchical information derived from CAD models is the basis for representing features. They can be divided into three levels [145]. The highest level are topological data that refer to information about the hierarchy and adjacency between pairs of vertices, edges, faces, etc. The next level is coarse geometry data that is regarding to the specific attributes of vertices, faces and edges (e.g. concave, convex, smooth). The lowest level is fine geometrical data such as angles, dimensions, tolerances, etc. The pattern of combining these three levels to represent features is what we call representation scheme. Precise choice and utilization of these information are the keys to the development of a feature recognizer, which lead to the emerging of various feature representation methods. [97] introduced the concept of implicit and explicit feature representations. For explicit representation, a full geometric shape needs to be defined, while in implicit representation, minimal constraints are collected to define the feature and other information can be derived if needed. For example, [35] defines a feature library with five geometric attributes: protrusion/depression, external access direction, exit status, boundary perimeter and boundary geometry.

From a CAD point of view, there are three types of representation schemes: 2D wireframe model, surface model, and volumetric model. Table 1 shows the comparisons of different types of representation schemes [115].

3.1 Wireframe Representation Scheme

Wireframe representation is composed of points, lines and curves that constitute the edge boundaries of parts. It is easy to use at low computational cost; however, it suffers from problems that are associated with informationally incomplete, ambiguity in solid information, limitation to complex shape. These problems also led to subsequent development of 3-D modelling.

Despite the limitations, feature recognition based on 2D wireframe models still has been explored since majority of existing design parts is in the form of 2D drawings in the old days. In order to handle these legacy product models, these approaches mostly were realized based on the research of objects reconstruction from orthographic projections [112]. For example, [85] demonstrated that it is possible to recognize features from 2D drawings by detecting the relationship between line segments and arcs. [75] used the divide-and-conquer strategy to extract the vertex-edge data from 2D engineering drawings. Then, a set of rules were developed to match the possible loop patterns of decomposed vertex-edge graphs with a feature library of orthographic projections. [120] deems the feature recognition techniques based on 3D model are complex, in consideration of the need of feature recognition for legacy drawings, they converted 2D drawings into 2.5D subparts first, then five geometric attributes defined in [35] are analyzed for isolated features and matched with a feature library. However, there are crucial limitations for 2D feature recognition methods, such as limited to simple, uniformly thick, specified and non-interacting features.

3.2 Surface and Volume Representation Schemes

Surface representation describes the feature by a collection of boundary faces (and possibly edges and vertices) that created by machining operations. It was designed to address the ambiguity problem and easier to interpret than wireframe models. They can be converted into graph structures including the faces and their connecting edges for further manipulations. Complete surface models also can be called solid models because they completely describe the connectivity of surfaces of the solid objects and how they form completely closed and connected volumes [19].

	Attributes	Wireframe model	Surface model	Solid model
1	Simplicity	Very simple	Complex	Highly complex
2	Utilization of computer time and memory	Least	Medium	Most
3	Ambiguity	Very ambiguous, relies on human interpretation	Less ambiguous than a wire frame model	Very less ambiguous
4	Users training	Comparatively easy	Requires more training and mathematical background	Knowledge of sections, and engineering drafting preferred
5	Types of data required for its construction	Both geometrical and topological	Only geometrical data	Only geometrical data
6	Applicability of shading algorithms	Not available	Available	Available
7	Storage of			
	geometrical data	Yes	Yes	Yes
	topological data	No	No	Yes
8	Ease of defining an object (e.g. curves, surfaces and solids)	Difficult	Easy	Easiest
9	Calculation of mass characteristics	No	No	Yes
10	Suitability for NC tool path generation	Medium	High	High
11	Capability to handle spatial addressability	Incapable	Incapable	Capable

Table 1: Relative comparisons of the basic types of model representations [115].

Volumetric representation describes an object as a set of primitive volumes. It was designed for extending feature concepts to general machining volumes that are associated with particular machining operations, because machining process is more easily described volumetrically. There are mainly of two types solid modelling techniques: Boundary Representation (B-Rep) and Constructive Solid Geometry (CSG). In a B-rep model, features can be represented in either of these two schemes as volumes or surfaces, whereas in a CSG model, features are always represented as volumes. Figure 7 shows a part together with the two types of feature representations, surface and volumetric scheme.

Volumetric representation can resolve certain feature interaction problems to some extent. Feature interactions may either create or destroy boundary elements, by which affect the features' topology more than their geometry. In a surface representation scheme, the topology variations are difficult to be eliminated, especially for entirely disappeared faces and edges. However, in volumetric representation scheme, interacting features can be considered as overlapped volumes. Therefore, by volume decomposition methods, the overlapped volumes can be combined with adjacent volumes in different ways to generate possible interpretations of feature interactions.

3.3 Boundary Representation (B-Rep)

B-Rep is an explicit representation of the solid; it describes the geometry and topology in terms of its boundaries, including the vertices, edges, and surfaces. These basic geometric entities are low-level information and provide convenience for recognition algorithms to act on, hence B-Rep is extensively used in the field of feature recognition [71].

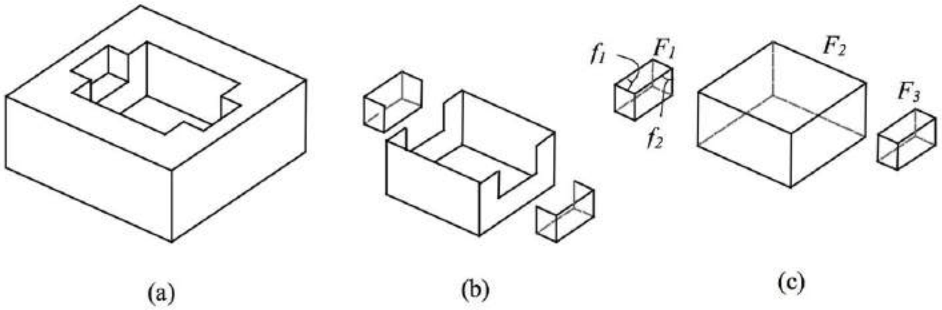


Figure 7: Surface and volumetric feature representation [142].

In a B-Rep model, vertices are the zero-dimensional entities. They are the intersecting boundaries of edges, carrying with the information of angle, connectivity and associativity between edges. Edges are one-dimensional entities. They constitute the wireframe of solid models and are the intersecting boundaries of two adjacent surfaces [28], carrying the information of curvature and convexity between faces. Faces are formed by a loop of edges. They form the border and envelop volumes of the part. The attributes of a face include the face properties (planar face/surface, machining face/stock face), the boundary enclosure box of the face, the closeness of the face and the adjacent faces [71]. The basic concept of convexity/concavity of the edges was first introduced by [62]. Convexity/concavity is a fundamental characteristic of polyhedral objects that allows some effective solutions of geometric problems. A complete description on convexity and concavity of faces, edges and loops in the solid model can be found in [141] and [28]. Figure 8 shows a simple example of different types of edges.

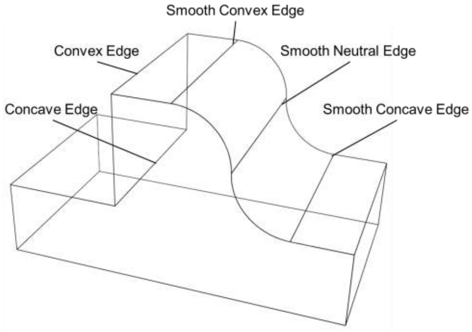


Figure 8: Convex, concave and neutral edges.

B-rep is an outstanding scheme with attractive properties such as applicability, sensitivity, and precision. However, it describes the parts at a consistent low level, resulting in the lack of conciseness and efficacy [19]. Moreover, B-rep is not the unique representation of shapes, since the boundaries of object can be grouped into different sets. In order to apply the feature notions, the primitive entities in B-rep have to be combined to compose an object representation with unique topological descriptions, even when the objects have different geometries. These problems led to the development of graph-based models. As shown in figure 9, a 3D solid model in a B-rep scheme can be considered as a graph structure [15]. Each node in the B-rep is labelled as numbered topological entities such as faces, edges, loops, or points. Each pointer denotes a hierarchical relationship.

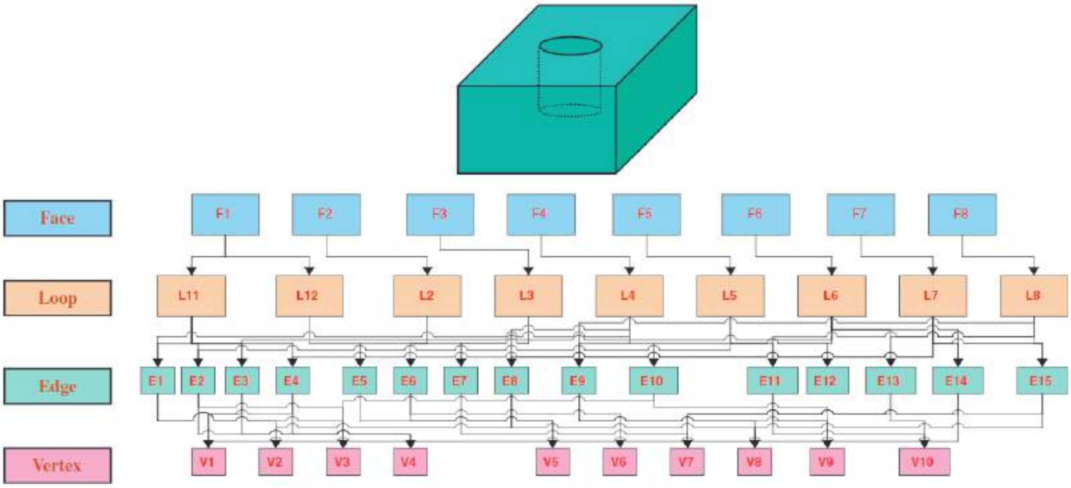


Figure 9: A boundary representation example of a blind hole feature [89].

3.4 Constructive Solid Geometry (CSG)

CSG models describes a part by Boolean expressions and rigid motions of solid primitives, and the topology and geometry are stored in an implicit way and need to be calculated from the set of solid primitives. The standard primitives are the parallelepiped (block), the triangular prism, the sphere, the cylinder, the cone and the torus [41]. [68] introduced the CSG tree, in which the whole part is represented by a tree whose nodes are the solid primitives and Boolean operations on these primitives. Figure 10 shows an example of CSG tree. One major advantage of CSG tree is that the nodes and features may be easily modified and arranged by order of construction or destruction [49]. However, CSG tree suffers from implicit representation and non-uniqueness, so it is not widely used in the feature recognition area.

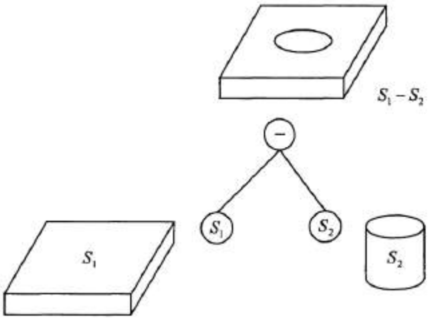


Figure 10: A CSG tree of a solid model [8].

3.5 Graph-based Representation Scheme

Due to the advantage of a clear separation between topological and geometric entities, boundary representation of the part can be conveniently transformed into graph-based representation scheme. In a graph-based model, the relationship among pairs of primitive topological entities are explicitly represented in a hierarchy of relational models, in which the root corresponds to the main object entity and other tree nodes provide a graph-based representation of deduced information [19] [25]. It allows a better interpretation of the topological structure of an object, also facilitates the development of feature recognition technology. Figure 11 draws a timeline of the development of the graph-based representation scheme.

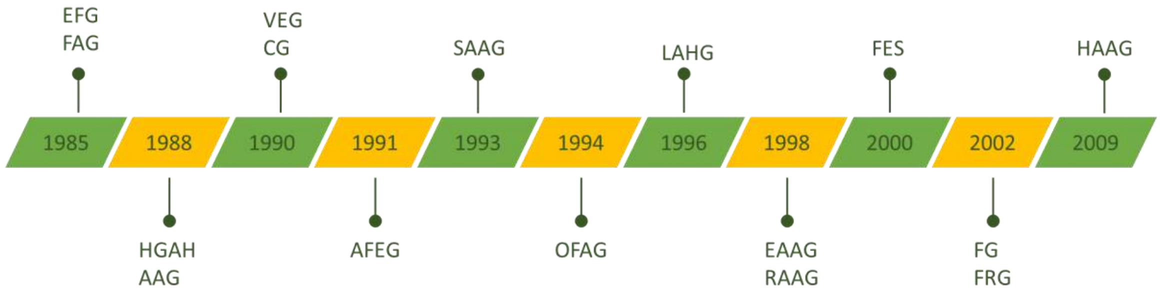


Figure 11: The timeline of graph representation development.

[4] proposed the edge-face graph (EFG) based on the adjacency between faces and edges. In EFG, the nodes represent faces, arcs, and edges of the corresponding object. The total graph explicitly represents the faces of an object and their mutual adjacency relations. [3] further developed EFG by adding a dashed arc representing vertices connectivity. The new graph was named face adjacency hyper-graph (FAG). It becomes a complete description of the hierarchical graph structure of an object, in which the nodes correspond to the object faces, and the arcs and dashed arcs represent relationships among faces induced by the edges and vertices. On the base of FAG, [19] developed hierarchical face adjacency hypergraph (HFAH), in which faces are clustered as shapes according to the adjacency relations and can be organized into a hierarchical form. However, this method was not capable of classifying complex and interacting features. In order to easily define features, [15] proposed a labelled graph representation scheme based on vertex-edge graph (VEG), in which both vertices and edges are labelled by convexity.

Among various graph representation schemes, one most popular is attributed adjacency graph (AAG), introduced by [51]. It is the basis for numerous subsequent approaches inclusively known as graph-based methods [101]. As shown in figure 12, AAG is a graph whose nodes are faces, arcs are edges corresponding to face adjacencies, and arc attributes account for the edge convexity. In the literature, AAG is then decomposed to its sub-graphs by removing all of its nodes surrounded by convex edges. The resulting sub-graphs are analyzed to determine their feature types with the aid of feature template graphs. This work only considered 2.5D polyhedral parts with predefined feature interactions, and it was computationally intensive. In addition, no face adjacency relationship was developed for interacting features. Nevertheless, it inspired a large amount of successive work.

To better support feature recognition, researchers keep looking for higher level of abstraction. [80] proposed the cavity graph, which is similar to the AAG but with the face node labelled by its orientation direction. They also introduced the concept of virtual link to help recognize interacting features. [17] introduced the concept of aspect vector (tool approach direction for machining the feature) to extend EFG and named the new graph as aspect face edge graph (AFEG). [63] extended AAG by adding curved surface node and developed surface-based attributed adjacency graph (SAAG). [27] developed oriented face adjacency graph (OFAG) based on FAG by adding two labels to the arcs, one indicates whether the adjacency is convex or concave, and the other describes whether the adjacency is interior or exterior with respect to one of the faces. [99] proposed a loop adjacency hyper graph (LAHG) where a node represents a loop and arcs denote the edges with attributes of convexity. This method focused on extracting the features from polyhedral parts by tracing their effect on changing the basic shape of an object.

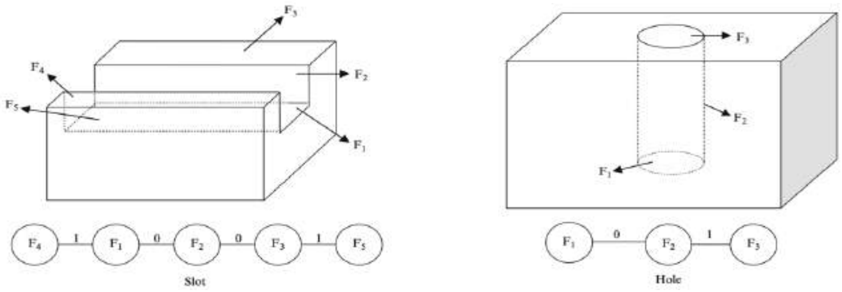


Figure 12: The AAG examples of (a) slot and (b) hole [54].

They classified features as surface, edge, vertex, mixed and global. [143] introduced the constraint satisfaction concept to graph matching and proposed constraint graph with perpendicularity, by which opposite and unifiable relationships between non-adjacent faces are defined in the part by labelling the arcs.

[122], [123], [145] developed the Multi-Attributed Adjacency Graph/Matrix (MAAG/MAAM) to overcome the limitations of AAG. In comparison with an AAG where only the arcs have attributes, both the nodes and arcs can be attributed in a MAAG. In addition, the number of attributes associated with the nodes or arcs is not limited. MAAM is a face adjacency matrix in which the diagonal cells contain the face attributes and the off-diagonal cells contain the edge attributes. In order to enhance the ability of interacting feature recognition, [146] modified AAG by adding the "reference face" attributes to both nodes and arcs and called it RAAG. Reference face was defined as faces with non-convex edges that indicates the presence of features. To better support interacting features, [30] extended AAG with more attributes of edge and face, namely extended attributed adjacency graph (EAAG). The added five attributes are convexity, existence, loop, geometry and blend type. Instead of building a complete graph for the whole part, [134] proposed an isolated version of AAG, which only considered the concave edges as hint of potential features, then attached the adjacent faces to the concave edges to construct the face edge sequence (FES) graph. The author also introduced a new kind of adjacency relation, the next-next edge, to provide more topological relationship between the edges. In order to support freedom surfaces and edge features, [73] improved EAAG by adding edge node type and quantitative attributes such as face normal vector, face angle, and edge length, and named it as holistic attribute adjacency graph (HAAG).

[42] developed a multilevel graph representation scheme representing different levels of abstraction of the geometry of a part to handle interacting features. The low-level graph was face graph (FG) similar to AAG, having faces as nodes, but with attributes could be coefficients of the equation of a plane or the position and radius for a spherical face. The high-level graph was feature relation graph (FRG) whose nodes were primitive features obtained by low-level feature recognition or from a feature-based CAD system, and the connecting arcs represent feature relationships, such as intersect, parallel, stackable and mergeable. Each node or arc of an FRG had a vector of attributes. The node attributes of FRG could be the feature's position, orientation and other parameters unique to that feature type.

The variations of these schemes can be understood that they are modifying the illustrated graph structure in two aspects [140]: (1) labelling nodes and arcs in different way to give them more attributes, or in other words, geometric constraints; (2) interpreting nodes and arcs in diverse sets, for example, faces and edges, surfaces and adjacency relationship, or vertices and edges respectively. These various graph representation schemes were developed to enhance the representation capability of graphical models for more and more complex features. The information stored in graphs gradually became more complete and less ambiguous, however, at the cost of robustness and computation complexity.

4 METHODS

Feature recognition is a long-evolved concept and has been active more than two decades in academic research works. In the field of feature recognition, many different techniques have been proposed. However, although many questions remain unresolved, the interest for feature recognition has subsided over the past 10 years, assumedly because the need to integrate CAD to CAM is decreasing along with the development of CAD and CAM platforms, since their integration has been a direct enabled technology for computer-aided technologies (CAX) system.

The basic problem in feature recognition system is the identification of high-level information, or in other words, finding implicit patterns from an object represented by explicit geometric entities. This problem can be formulated as a geometric constraint satisfaction problem [143], in which variables represent entities, the constraints are expected geometrical and topological properties of the variables, existing solutions are either finding determined values or generated classifications for the variables. Therefore, in most proposed methods, this problem is solved by matching the rules, graphs or templates, i.e. constraints. And the recognition process can be generally divided into two steps: decomposing the object into low-level representing entities and applying the reasoning process to classify set of entities as certain features.

There are numerous systems developed by attempted work based on the pioneer work of [62]. Some of them either have not been fully developed or have been replaced by newer techniques that have overcome their limitations. Therefore, this paper focuses on the four approaches attracted most extensive research interest and the hybrid systems combing their strengths: rule/hint-based approach, graph-based approach, volumetric decomposition approach and artificial neural network approach. These approaches are presented in the following sections.

4.1 Rule-based and Hint-based Approach

The rule-based approach was among the earliest to be investigated due to the granted advantages of expert system, such as [10], [23], [43] and [128]. Features are generalized as templates consisting of characteristic patterns of rules, but no explicit representation scheme was defined for feature extraction. The recognizing process is performed using these inference rules as If-Then. If the predefined conditions are satisfied, then the corresponding structure in the part is recognized as a feature.

Rule-based approach is straightforward and easy to build; nonetheless, it also has many fatal weaknesses. On one hand, since the feature representation is ambiguous while the rules have to be predefined, rule-based systems are inflexible to be scaled up. On the other hand, not every complex feature and interacting features can be defined by rules. Given more complex conditions, the system becomes more difficult to conceive, as less uniqueness remaining, and less likely to successfully implement rule recognitions. In fact, the main idea of this approach is setting up boundary constraints of features. The constraints don’t need to be generalized rules nor demonstrate engineering significances. Hence, rule-based was later developed into hint-based and graph-based methods.

In order to deal with feature interactions and be more flexible with the feature searching, hint-based methods were developed based on the idea that incomplete representation can be searched for indication of the existence of certain features. Because exact patterns/rules searching is very likely to fail when features shift or intersect. Hence, a hint can be defined as a pattern in the part boundary that provides a trace for the potential existence of a feature [49]. This method was initiated by [121], in a system called Object Oriented Feature Finder (OOFF). In the literature, the concept of hint was derived from the “presence rule” which is a minimal indispensable portion of a feature's boundary which must be present in the part even when features intersect. A hint generating strategy was firstly proposed as combination of part faces, which contain characteristic information from diverse sources that can be associated with a certain feature type. This system was further improved by [40] and [39] by providing it with the ability to reason about hints generated from various sources, which may include direct user input, tolerances and attributes, and design features.

Hint-based approaches use a two-step procedure to feature recognition; in the first step, hints are generated by extraction rules based on different attributes, such as geometric and topologic reasoning [7] [20] [29] [125], feature taxonomies [28] [89] [141], and combined probabilities of ranking potential feature hints [40] [121]. In the second step, these hints are processed and may be directly matched by applying rules [7] [28] [89], while some works also have a test phase after constructing feature volumes from hints and boundary data [40] [121].

Although most hint-based methods define hints based on face patterns, there are also some attempts to enhance the methods by utilizing additional information on other geometric entities as hints. For example, [34] and [104] used edges and vertices rather than faces as hints to identify or decompose interacting features, then possible volumetric features are reconstructed to facilitate sequencing process in the CAPP system. In order to provide prompt manufacturing feedback at design stage, [49] extracted feature hints from wire-frame models for simplicity. However, as the disadvantage of wire-frame model mentioned in section 3.1, this method is not suitable for complex parts and interacting features. [147] consulted the part's CNC program to extract the feature information and retrieve process knowledge. [114] presented a novel methodology extracting hints by projecting and measuring virtual rays to objects. This technique is also known as "a viewer-centered approach" that mimics the way in which humans might observe and identify the faces and volume information of objects. [103] presented a similar method that gets the contour of 2.5D machining part's top view as feature hint by projecting virtual rays vertically from upside. Then the faces and volume information are extracted by analyzing the boundaries and length of the rays.

An observation that could be made from these studies is that the key in developing a hint-based system is how to design an appropriate hint extraction strategy. In particular, determining the characteristics of a feature presented in a part and maintaining its consistency in spite of feature interactions has been a difficult task. To tackle this, [9] proposed a method for automatic generation of logics as feature hints. These rules are formed by applying an inductive learning algorithm on training data consisting of sample features. First, sampled features are converted into characteristic vectors with the numbers representing the attributes of each face. Then inductive learning algorithm is applied to extract feature hints formed by subsets of rules, which represents patterns associated with individual feature faces. This method is also able to facilitate the expansion of existing feature library. However, how to apply these hints to recognize features and how practical this approach could be for complex interacting features remain unanswered.

4.2 Graph-based Approach

The Graph-based approach is among the most researched methods due to the inherent advantage of a graph's structural similarity with B-Rep based solid models. In graph representation schemes, nodes and arcs normally represent faces and edges, which are attached with some attributes such as the convexity and concavity of edges, type of face, perpendicularity, parallelism or tangency of edges and faces, etc. Then features would be extracted as subgraph from the complete graph. In various graph-based representation schemes, the features are typically predefined templates and constrained by three types of information: the required number of faces for composing a target feature, topological and geometric relationships among composing faces. The feature searching can be realized by subgraph isomorphic matching.

The approach developed by [51] is considered as the first formal graph-based feature recognition method, introducing the concept of an attributed adjacency graph (AAG). AAG captures the concave/convex relationships of the part's adjacent faces and analyze the adjacency graph in order to decompose it into subgraphs as features. Although original AAG concept is restricted to negative polyhedral objects, it inspired a large amount of successive work. As discussed in section 3.5, the main development trend of graph-based approaches was to enhance the capability of graph-based systems by enriching the expressiveness of the feature graph with more attributes.

Besides representation scheme, the other key variation of various approaches is feature extraction process, or in some cases, graph decomposition. Graph isomorphic matching is the basis for all graph-based methods. In early works, such as [15] [27] [110], the primitive templates were compared to the entire graph one node by one node. This exhaustive matching is a well-known NP-hard problem and requires all N! reordering and comparisons [124]. In order to reduce search space and support interacting features, the further studies introduced graph decomposition method.

The well-known method proposed by [51] decomposed the part graph into subgraph by deleting nodes representing faces that were connected to all adjacent faces with convex angles. This is based on the observation that such faces generated by part of machined features. Thereafter these subgraphs were matched with templates for recognition. [17] [18] and [32] assumed that all features/subgraphs were bi-connected and tri-connected nodes. Therefore, the graph was partitioned at cut-nodes, which would be identified first using different algorithms. Since the cut-node is considered as the entrance face of a depression feature, this method cannot recognize features with more than one entrance face. [80] disconnected the nodes having only concave edges, namely using cavity graph as subgraph, which represents depression features. [63] introduced an incremental recognition method, where the entire graph was matched with a shape to determine the subgraphs for further matching. [123] transformed the part graph into a Multi-Attributed Adjacency Matrix, where each row and column represented faces and the cells were the adjacency indicated by numbers. In the step of feature extraction, the matrix was scanned to classify concave and convex faces as root and boundary faces respectively to form a feature/vector. [99] used loops as the descriptive unit for object representation. Loops are defined as cyclic sets of edges, and classified into convex, concave or smooth types. In this work, features were extracted according to the specific loop types. [146] assumed that the non-convex edges of a non-convex hull face are connected to some feature faces. Therefore, the faces contained non-convex edges were used as reference for the presence of features.

[30] introduced the concept of Minimal Condition Sub-Graph (MCSG), which was interpreted as a hint of potential feature existence in element construction. To generate MCSG, first split the part's EAAG into separate manufacturing face adjacency graphs (MFAG) by deleting all the stock face nodes and their incident arcs. The MFAG was compared with the predefined feature library, if not matched, it would be regarded as interacting feature and further decomposed into partly concave adjacency graphs (PCAG). Then each PCAG of the MFAG was checked to see if it was a concave adjacency graph (CAG) or general feature. The concave adjacency graph (CAG) is a connected subgraph where all arcs are concave. If true, this PCAG was a MCSG. If not, this PCAG would be decomposed into separate CAGs by splitting the shared faces caused by interactions, so as to get all MCSGs. Finally, MCSGs are completed to a recognizable form by restoring their missed links.

[145] presented a method called concave triggering algorithm (CvTA) to extract sub-graph/feature from the MAAG, which was designed based on the principle that a segmentation occurs whenever there is a concave edge. [42] developed a multi-step graph matching algorithm based on a multilevel graph representation scheme. The matching starts from only one vertex, namely a seed match, which then is extended to adjacent vertices and edges. Finally, the feature candidates would be matched with respect to both topology and constraints. Based on MAAM, [124] introduced a concept of degree of vertex, which was calculated by the number of incident edges to the vertex. In fact, the degree of vertex indicates how many concave edges it has. According to the degree of vertices, the MAAM would be partitioned and rearranged into several sub-matrices, which could be further matched with the predefined feature vectors. [73] predefined a library of feature seed faces that combines the key faces with their edges that can mostly reflect feature characteristics for each type of feature. These seed faces were used as hints for searching in the part graph. Then the discovered hints would be extended and combined with features based on rules.

It can be stated that graph-based approaches are quite effective in the domain of isolated features, which are the features without any topology variation caused by interference with other features. However, the most significant drawback of all graph-based methods is their incapability of handling arbitrary feature interactions. This is because feature interactions can destroy topology of basic features and hence cause the missing arcs in the graph.

There are some attempts, such as [1],[46],[47],[79], to directly restore the missing arcs using a concept of virtual links, which are identified from a set of possible candidates created by extending specific faces. The candidate links are ranked based on the geometric and topological evidences at different abstraction levels using Dumpster's rule of combination, and then the highest ranked one is restored as missing arc. Nonetheless, the problem was not solved satisfactorily, as the ambiguity of feature combinations could not be eliminated. It is even worse when vital faces are destroyed completely that it can't to be restored by extending residue faces. In general, this method has fatal weakness to get desired missing arcs for proper decomposition of delta volume.

The other solutions to handle missing arcs are predefining some interacting features as high-level feature [42] or decomposing certain types of interacting graph into subgraphs by heuristics [30]. Although it is unrealistic to enumerate the unlimited types of feature interaction, these predefined classes can be recognized at least for specific post-processing applications. However, in this way, the feature library could not be expanded automatically, and the imposed design constraints would limit the feature template to essential shapes. The small-scale variations such as fillets and chamfers, need to be removed before graph construction. Due to the combinatorial difficulties and exponential time complexity, with more small-scale details and complex shapes, more ambiguity as well as computational time in recognition process. Consequently, graph-based approaches have difficulties to handle real industrial parts.

From all these shortcomings, many alternative approaches, such as volumetric and hint-based methods, were investigated to deal with interacting features. However, there still have been a lot of attempts to integrate graph-based methods into emerging approaches, on account of its flexibility to expand feature library without changing recognition algorithm and excellent ability to extract feature candidates from solid models.

4.3 Volume Decomposition Approach

The volume decomposition approach identifies the removed (machining) volume of material from solid stock and decomposes the volume into intermediate volumes first, and then the features are generated by combining the intermediate volumes based on certain rules. According to the methods to decompose volumes, volume decomposition methods can be generally divided into two sub-groups: convex-hull decomposition and cell-based volume decomposition.

4.3.1 Convex-hull decomposition

The technique to decompose non-convex objects into convex components with arbitrary shapes was introduced by [11]. For machining purposes, [135] implemented it to express a non-convex object in the form of a sequence of convex volumes called Alternating Sum of Volumes (ASV) Decomposition. As shown in figure 13(a), the first step is to determine the part's polyhedron convex hull. Thereafter the volumetric difference between the part and its convex hull is computed recursively, until the convex hull equals the part. The Boolean combination of resulting convex volume was defined as an alternating sum of volumes (ASV). To suit various manufacturing environments, the ASV expression can be algebraically reformed into disjunctive normal forms. As shown in figure 13(b), the most underlined problem of ASV decomposition is that the recursive decomposition may not terminate due to non-convergency.

[118] investigated the causes of this non-convergence, and found that the vertices not on the boundary of convex hull would lead to the non-convergence. The authors defined these vertices as non-supportable vertices and designed a detection algorithm. Therefore, the solution to non-

convergence is proposed as that once occurs, the problematic volume would be separated into pieces around the non-supportable vertex.

[60] did similar research about the non-convergence and gave a more complete definition of non-supportable vertex. To solve this, [59] proposed the method of Alternative Sum of Volumes with Partitioning (ASVP). In ASVP, they applied ASV first until non-convergence was detected, the delta volume was partitioned by cutting along planes spanned by a pair of incident edges to the problematic vertex. Then ASV decomposition was applied to each piece separately until another non-convergence occurred.

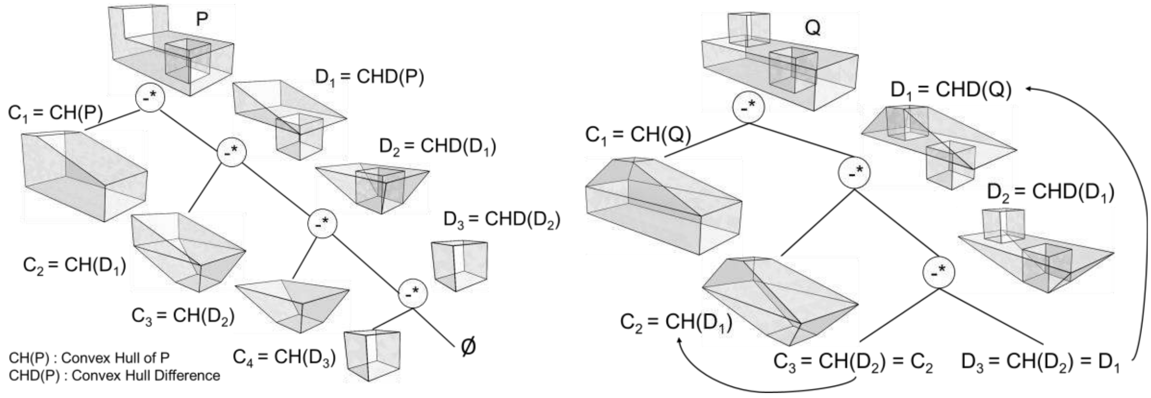


Figure 13: (a) ASV decomposition, (b) Non-convergence of ASV decomposition [57].

Figure 14(a) shows an ASVP feature recognition system developed by [57] based on the observation that since each ASVP component represents partial boundary information of the given object, ASVP decomposition can be considered as a hierarchical volumetric representation of form-features. Therefore, as shown in figure 14(b), feature can be recognized by finding intrinsic interrelations between the object faces contained in the decomposed components according to the hierarchical structure of the decomposition. If there are two or more connected original faces in an ASVP component, it can be recognized as generic feature based on the volume contribution and the normal vectors of the original faces. For the unrecognized components, the author introduced a method of immediate super-component (ISC) combination of original faces with the corresponding components on the basis of the hierarchical structure and face-dependency information of the decomposition. The resulting volumes were represented by a binary tree comprising Boolean operators such as union and subtraction of the decomposed volumes. This method is also called form-feature decomposition (FFD).

Since ASV/ASVP decompositions are purely based on geometry, the shape of FFD is arbitrary and not corresponding to any manufacturing operation. Therefore, to generate manufacturing features, a post-processing module is needed. In fact, FFD maintains enough information of geometry and adjacency to support the geometric reasoning of various manufacturing and design activities. [129] proposed a method to convert FFD into machining features by rewriting the Boolean expression of every positive form feature using the half-spaces determined by its original faces. [93] improved ASVP method by adding incremental update capability to support concurrent feedback of design changes. This was achieved by incrementally updating the corresponding ASVP decomposition and applying combination operations to the updated portions of the ASVP decomposition.

[58] developed a feature recognition system using alternative sum of volumes with partitioning decomposition. The objective of this method was to systematically obtain geometry-based relations between machining feature recognized from the part boundary. The precedence relations

between features are achieved by combining the face dependency information with machining process information. The precedence relations for a given feature decomposition are then recognized as a set of precedence trees. In addition, each precedence tree for a given feature decomposition represents a different set of the features. [26] also described a convex-hull-based method for 2.5D prismatic parts. Features are recognized by identifying the profile patterns of convex-hull, based on the observation that the inner loops of concave edges lying within the convex hull imply machining features.

Although convex-hull decomposition methods are effective in finding delta volumes for polyhedral parts, they have difficulty with curved surfaces. For instance, [87] described an ASVP implementation on integration of CAD and CAM for 2.5 and 3-axis machining centers. However, it cannot be extended to freeform surface because it is difficult to define the convex hull of a curved surface. [83] and [24] attempted to include cylindrical surfaces into convex-hull approaches by assuming the convex hull of cylindrical surfaces are still cylindrical, but they can only handle limited cases of face intersections.

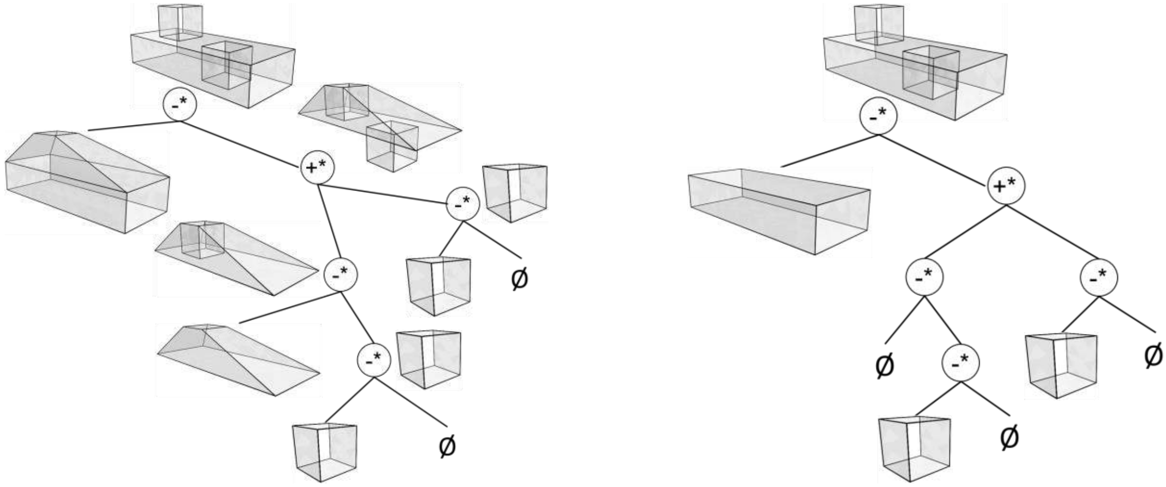


Figure 14: (a) ASV decomposition, (b) form-feature decomposition [129].

4.3.2 Cell-based Decomposition

The essential methodology of cell-based decomposition approaches is to decompose the volume or delta volume of an object into minimal cells with a simple shape. The cells are then recombined as a larger volume that can be removed by a single machining operation, and in the last step, the combined volume is checked for topological and geometrical characteristics and recognized as a machining feature. The variations between the proposed cell-based algorithms mainly lie in how to decompose and recombine cells into recognizable volumetric features. Moreover, the difficulty is how to generate suitable feature sets efficiently.

[33] was the earliest example of cell-based method. This work developed an algorithm to decompose delta volume of a part into a set of generic shapes, which would be combined next by face extension and sectioning. Then the combinations are matched with predefined library, but the recognition sometimes fails due to feature interactions.

As shown in figure 15, Sakurai and Chin [108] proposed a representative method to generate all the possible combination sets of the minimal convex cells decomposed by face extension. Due to the exhaustive nature of this method, the opportunity of recognizing all the features correctly is guaranteed. However, generating these combinations is very computationally expensive, especially if curved surfaces are included. Even though some checking rules were set to discard some unpromising sets, the process is still too verbose even for a somewhat complex part. [119]

developed a similar method that decomposed entire delta volume of part into blocks by extending bounding faces. Then the blocks are connected systematically in different ways to generate multiple interpretations of feature. By using graph-based methods, the reconstructed volumes are matched with the predefined features and classified accordingly. The drawback of this method is that it is only applicable to the features whose faces are orthogonal or parallel to each other.

In [113], Shah decomposed the part into minimum convex cells first by half space partitioning method. Then the cells were reconstituted into maximum convex volume by the aid of a cell adjacency graph. Since concatenated volumes are not overlapped and regarded as machining features, they can be classified with respect to machining attributes such as accessibility and degree of freedom. [16] developed another analytic face extension method to decompose an object into minimal cells called base volumes that compose only convex maximal features. Then specific types of feature are recognized by graph matching on the maximal simple features.

[107] presented an improved version of their earlier work. To solve the problem of unreasonable composition and to avoid recognizing overlapped features, they introduced a more efficient decomposition method based on half space partition and a maximal convex cell method. After generation, these cells subtracted from each other in different orders to yield multiple interpretations of features in delta volume. However, the total number of interpretations increases exponentially as the number of cells increases, and unnecessary interpretations problem still exists. In addition, this paper did not give much detail about how to combine these intermediate volumes into machining feature, nor the feature recognition process.

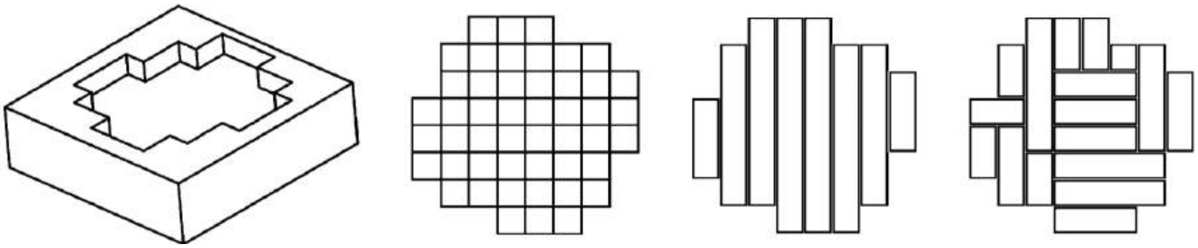


Figure 15: Cell decomposition method illustration: (a) part; (b) cells; (c) a reasonable set of cell combinations; (d) an unpromising combination result [6].

In [109], Sakurai and Dave improved the cell decomposition method in their earlier works, and successfully applied it to the object with curved surfaces. The method still decomposes an object into minimal cells by extending faces or half space of the delta volume. In contrast to other methods, it allows composed volumes to be concave, namely maximal volumes, within the half-spaces of object. The combinations of minimal cells produce maximal volumes by examining the relationships among these minimal cells. The author stated that the maximal volumes would be recognized by graph matching. [88] and [74] adopted similar delta-volume decomposition method to slice the entire delta volume along a certain axis into sub-volumes having constant cross-section. Then the sub-volumes are decomposed again into multiple sets of machine-able volume, which is considered as machining feature.

[53] introduced a concept of machined face, which is the face of part at which material is removed by machining processes from the stock. It can be found by comparing the final part and stock. After identification of machined faces, the delta volume is decomposed recursively by extending the machined faces, until there is only one machined face in the partitioned volumes. Then the resultant volumes are reconstructed and mapped to multiple interpretations of machining operations.

[138] introduced a new cell decomposition and combination method based on the previous work in [109]. As the example shown in figure 16, the new method decomposes the delta volume

by recursively bisecting sub-volumes into two smaller volumes with similar number of faces until each volume has less than 16 faces. The bisecting planes are recorded and used as reference to recompose these small volumes into maximal volumes which don't have concave edges and are not contained one in another. Then the selection of a set of non-redundant maximal volumes is performed by choosing maximal volumes containing faces of delta volume uniquely and having largest number of delta volume faces that are not contained in already selected ones. In the last step, maximal volumes are recognized as maximal features if they can be produced by a single machining operation, otherwise they are further decomposed to qualified shapes. This method significantly reduced the computational load of cell-based approach by having a smaller number of cells, so that also improved its applicability to complex parts.

In [136], Woo continued to improve his method in the spirt of reducing computational complexity by avoiding generation of large number of unnecessary cells. He described a concept of "localized face extension" that not only extends a single face of part but also connects it with the adjacent faces sharing the non-concave edges first, then the connected face is extended to intersect with the original part model and get new faces with edges. If a new face has concave edges, it is selected to be untied to form a cell. In the stage of cell reconstitution, the author defined a type of "seed cells" based on the fact that they always exist in maximal volumes. By using seed cells to merge adjacent cells under certain rules, the numbers of possible interpretations are significantly reduced.

[61] proposed a wrap-around decomposition method, in which wrapping an object using plastic wrap has been imitated. The wrap-around operation plays a similar role in finding a convex hull in convex decomposition. In wrapping operations, a convex inner loop is used as a clue to find a concave volume, which is decomposed by removing the convex inner loop and internal faces not constituting the convex volume. This decomposition method is efficient but only considering simple concave spaces bounded by a convex inner loop.

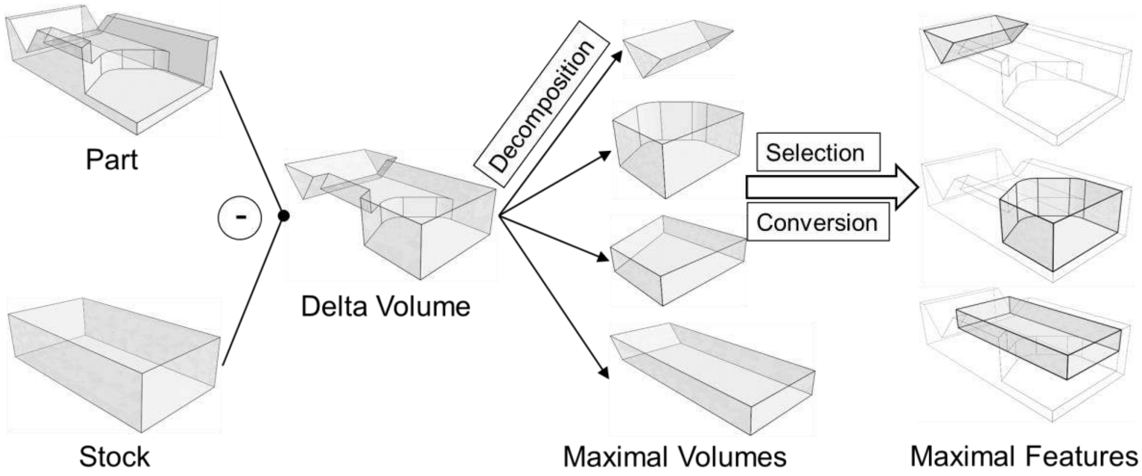


Figure 16: Maximal volume decomposition method [138].

[55] and [56] integrated the wrap-around decomposition with another three volume decomposition methods (fillet-round-chamfer decomposition, volume split decomposition and maximal volume decomposition). B-rep models are first simplified by a fillet-round-chamfer decomposition method. The volume split decomposition is used to split volumes by concave inner loops. The wrap-around and volume split decomposition methods are applied recursively until the sub-volumes are not decomposable. Then the maximal volume decomposition from [137] is adopted to generate maximal volumes. Finally, a volume decomposition tree that consists of Boolean operators such as

unions and subtractions for the decomposed volumes is generated for downstream applications. Comparing with other cell-based methods, the decomposition tree provides more hierarchical feature information. In addition, by using the wrap-around and volume split decomposition, this method reduced the complexity of cells regarding their number and shape so that the computational load is alleviated as well.

The features defined in cell-based decomposition approach are essentially volumes having simple shapes, which can be produced by a single machining operation. Therefore, this approach is suitable for manufacturing planning and NC coding. Other examples of applicable problems are such as mesh generation [77], model simplification [56], and feature-based model modification [55]. Moreover, because the exhaustive multiple interpretations enable the extraction of all features, this method has been considered as an effective method for handling feature interactions to some extent. These multiple interpretations can be used as a basis to generate alternative process plans, and potentially used in various downstream engineering activities, such as a design analysis and optimization.

However, the volume feature representation is deficient with geometric and topological feature information, therefore it is not able to recognize the feature types and hence not suitable for certain engineering applications. There are also some other disadvantages such as expensive computational loads and incapability of always generating features of interest. In some cases, the resultant volumes may not be even machinable features, thus need to be further processed in the context of machining to extract features of interest. In addition, because the decomposition process is based on face extension and edge convexity, the solid models having freeform faces cannot be handled properly.

4.4 Artificial Neural Network Approach

An artificial neural network (ANN) is a system roughly imitating human perception, which can intuitively process input information and output recognized results. The major characteristic of ANN that makes it one of the most promising feature recognition methods is its capability to derive implicit patterns through training with examples. In comparison with conventional feature recognition methods, ANNs do not implement any explicit reasoning operations. However, by simply performing arithmetic operations only, they can derive many kinds of knowledge and discover regularities through training implicit input patterns that are difficult to describe adequately with knowledge-based systems. The second interesting advantage for employing ANN in feature recognition is the robustness that can tolerant exceptions or incomplete input patterns, so it becomes possible to recognize non-orthogonal interacting features. To compare and evaluate different ANN-based approaches, special attention will be paid to the key factors of ANN functionality: the input representation and neural network architecture.

4.4.1 Neural Network Architecture

ANN's performance of learning and recognition is closely related to their architecture design, which mainly reflects selected training algorithms, design of network layers and number of neurons in each layer. However, there is no consensus among researches regarding which is the most suitable network for feature recognition. The network architecture is mostly designed through a large amount of experiments and evaluations with different combinations of parameters. In the existing ANN-based feature recognition systems, there are mainly three types of network architecture: feedforward neural network, self-organizing map and recurrent neural network. More detailed reviews can be found in [5], [21].

4.4.1.1 Feed-forward networks

Feed-forward (FF) neural network with backpropagation is the most often used ANN structure for feature recognition. It has an input layer, an output layer and one or more hidden layers. The number of neurons in the input layer is decided by the design of input representation scheme, while the number of hidden layers and the number of neurons in each hidden layer can be determined by trials and comparisons. In this type of network, neurons are fully connected across layers with varying weights and bias representing connectivity strengths. The network is trained by supervised learning, which is performed by adjusting these weights and bias to provide an effective mapping function between input and output vectors. Despite its relatively simple structure, remarkable learning and generalization capabilities are demonstrated for pattern association and recognition problems. Hence, three-layer feed-forward neural network has been used as classifier by many researchers to prove the feasibility of their methods, such as [12],[22],[52],[81],[72],[134].

Although three-layer FF network are able to approximate any continuous function, it needs too many neurons within a single hidden layer, which is prone to overfitting. Therefore, to deal with large number of feature types or complex interactions, one hidden layer might be not sufficient. For example, [92] and [117] adopted a standard four-layer FF neural network with two hidden layers due to its better convergence results. [90] presented a different four-layer FF neural network. A threshold layer is attached to the end of a three-layer network after training process, it filters out the output values smaller than 0.5.

In addition, by using genetic algorithm to find the best combination of effective variables in the network, [91] simplified the topology of their network in [92] for less computational complexity. Specifically, the network was successfully reduced from 10 input neurons to 7 and from 2 hidden layers each with 9 neurons to 1 hidden layer with 7 neurons each. The reduced network obtained better experimental results in terms of training time, processing time and computational complexity, and the accuracy of recognition was not affected.

[139] developed a network structure by cascading several typical three-layer feed-forward back-propagation neural networks. Each network is independently trained for a specific class of polygon feature. In the process of classifying a new input, the trained networks would be performed in sequence. If the new input is classified at a network, the classification terminates; otherwise, it is sent to the next network. If the classification fails, the new feature is stored for future training of a new class when the number of unclassified polygons reaches a preset value. [14] also adopted a cascaded structure consisting of expendable number of typical three-layer back-propagation FF neural networks. The most attractive advantage of this cascaded architecture is that it is easy to expend the domain of classification.

[95] developed a 5-layer FF network, functioning like a multilayer perceptron. The first layer converts input vectors into N integers, where N is the number of faces in the part. The second layer matches the integers with the M conditions required for the feature definition. The third layer collects the satisfied faces. The fourth layer verifies the collected face with the M conditions again. The last layer contains only one node, which is active only if all the M nodes in fourth layer are active. For this reason, the major disadvantage of this method is that the features still need to be predefined. [44] adopted the same ANN structure but with a different input vector representation.

4.4.1.2 Self-organizing neural network

Self-organizing networks cluster inputs by unsupervised learning. Such networks can discover similarities and correlations in the input data without reference to desired outputs and learn the patterns by adjusting the connectivity weights and bias. When a new feature is given to a network, if it is similar to one of the existing clusters, the corresponding cluster associates and adapts to it. Otherwise, the network memorizes it as a new cluster. The degree of patterns similarity within same cluster can be controlled by preset parameters.

Considering the outstanding ability to learn new patterns without any effect to previous learning, [66] presented an adaptive resonance theory (ART2) network to recognize machining features. To correlate between inputs and existing clusters, the similarity of the features is controlled by a reset mechanism via a vigilance parameter, which is critical to the network performance. The effect of the changes in the vigilance and the noise suppression parameters on the network performance were investigated, and the results show its potential applicability to feature recognition problems.

[86],[149] proposed a two-stage method to deal with interacting non-orthogonal features. In the first stage, the Kohonen self-organizing feature map (SOFM) is used to cluster vertices representing material volumes. Then the part's delta volume is decomposed into primitive features by Boolean operations to the resultant clusters of SOFM. In the second stage, the primitive features are recognized by a three-layer FF neural network described in [150]. The shortcomings of this work are that only simple interactions and only one interaction interpretation are considered.

4.4.1.3 Recurrent Neural Network

[130] proposed a two-hierarchy Hopfield network, which constructs a "Hypothesis -Evidence" system. The Hopfield network is a form of recurrent artificial neural network. It uses only one layer with binary threshold nodes for both input and output. The first network is used to generate possible hypotheses that matches the outline shape of input object with training samples. In the second network, the input object and sample model in the hypotheses are computed for normalized quantitative representation first. Then the hypotheses are evidenced to provide a correct match or not.

4.4.2 Input representation

Building an input representation scheme is the most important part of employing ANN in feature recognition. Manufacturing features can be characterized by both topological and geometrical information derived from the CAD model, while neural networks typically take numerical values as input. Therefore, this raises a problem of how to convert a solid model to a suitable input representation for neural networks, since simple numerical inputs are not always sufficient to fully represent geometrical and topological data stored in CAD models.

The reviewed input representations can be generalized into three types: 2D projections based, graph-based, and face score vector. To summarize their characteristics, a satisfactory input representation should at least have the following basic requirements: 1. it contains all the necessary information to identify patterns; 2. it must be unambiguous, which means each class should have unique representation; 3. it is implementable for computational devices.

4.4.2.1 2D-projection-based representation

[94] proposed an ordered triplet (C_i, A_i, L_i) to represent each curve segment of a connected profile in the 2D drawings, where C_i , A_i , and L_i are the curvature, interior angle and arc length of the i-th element respectively. An encoded feature vector of the triplet (C_i, A_i, L_i) for a given profile is used as the input of a three-layer feed-forward neural network. Later, [12] described a similar method to transform a 2D drawing into an ordered list of n line segments. Each line is represented by 7-tuple in the form: (L_i, A_i, C_i, J_i, OL_i, OA_i, OC_i) where L_i, A_i and C_i are the length, interior angle and curvature respectively. J_i is the line intersection type between the line segment and its subsequent line segment. OL_i, OA_i and OC_i are the ordinal values assigned to L_i, A_i and C_i that are used to find their magnitudes in case of size normalization. The line segment having largest size within a part whose OL_i is assigned to be 0th, the next largest is assigned first, and so forth.

[130] defined a pair of matrixes for representing the part's topology and geometry information respectively. The topological input matrix contains face adjacency, distance between face centroids, and shape rate. The geometric input matrix includes face type, face area, number of edges composing a face, and angle of vertex. The main problem of this method is that face orientations would affect the result of recognition.

[139] used a skeleton approach to simplify 2D projection views of a 3D prismatic part to get a tree structure consisting of line segments. Then tree-structured skeletons are converted into column vectors, in which each element corresponds to a line segment associated with six attributes depicted by numerical values. In addition, based on the number of successful

classifications with three different projection views, the classification results can be grouped by three different levels.

[14] deemed that the general appearance of an object is important for the classification, and only projected view contours is not enough to describe it, geometric and topological information also should be included. Hence, in their representation scheme, the part is projected along front/back, left/right, and top/ bottom views to gets nine different views, six of which are formed by the projected contour lines and visible edge segments while the rest three consist of contour lines and invisible line segments. As shown in figure 17, the visible and invisible projected edge segments partition each contour into different sets of regions. Then each view is represented by a graph in which the nodes represent the regions and the arcs denote adjacency between regions. Moreover, the edges of each region are coded into a representative ring code by clock-wisely travelling from a random selected edge. The code for each edge is determined by its traveling direction and a two-layer octal coding system. In the second step, the graphs are transformed into reference trees based on the weighting values computed with the representative ring code. In the last step, each reference tree node is associated with a number of values first. These values are derived from the representative ring code, region boundary and region index information. Finally, a vector form is generated with these values.

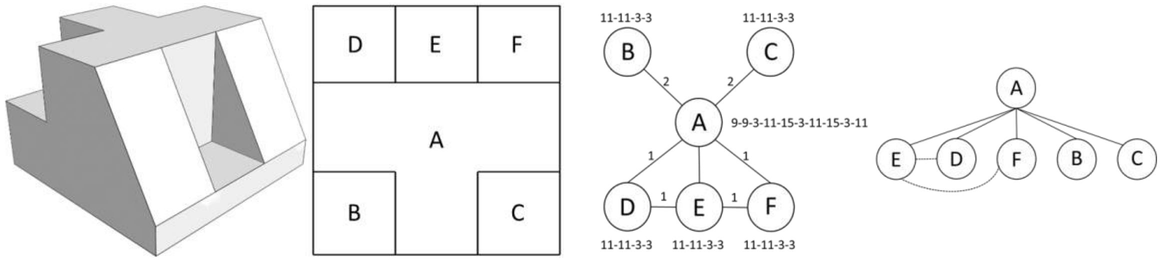


Figure 17: A 2D-projection-based representation scheme: (a) A 3D part, (b) the top view contour, (c) the weighted graph with representative ring code, (d) the reference tree [14].

[149] directly input 2D coordinates of vertex points in the top view boundary of part's delta volume to a self-organizing neural network. The vertex points are clustered into groups by neural network for interacting volume decomposition. However, this method only considered the cases of machining features interacting with uniform depths and common bottoms. In their companion paper [150], they proposed a different contour-based representation for forward feed neural network. As shown in figure 18, the top view contour is converted into a 3×3 matrix, where the four corner elements correspond to the four corner vertices, the center element indicates whether the feature is a solid or cavity, and the rest four elements represent the four edges. The matrix is assigned by numerical values indicating the types of vertices and edges. The advantages of this representation scheme are its flexible expansion and simple implementation. While the major limitation is that the input features are required to be defined by four rectangular vertices.

[52] proposed a "chain code" vector representation scheme for six types of primitive features, including block, hole, pocket, boss, step and slot. The number of elements in vector is fixed to ten. The first seven elements correspond to the approximate orientation of the line segments in the section profile. The rest three elements are assigned by binary values indicating the profile's attributes: convex/concave, circular /rectangular, and open/close. These three attributes are also obtained from vector manipulations.

In the comparison of 3D models, 2D projections are easier to be converted into numerical vectors. However, the possible line intersections and ambiguity of 2D projections would limit these methods to be applied to parts with simple block shapes only.

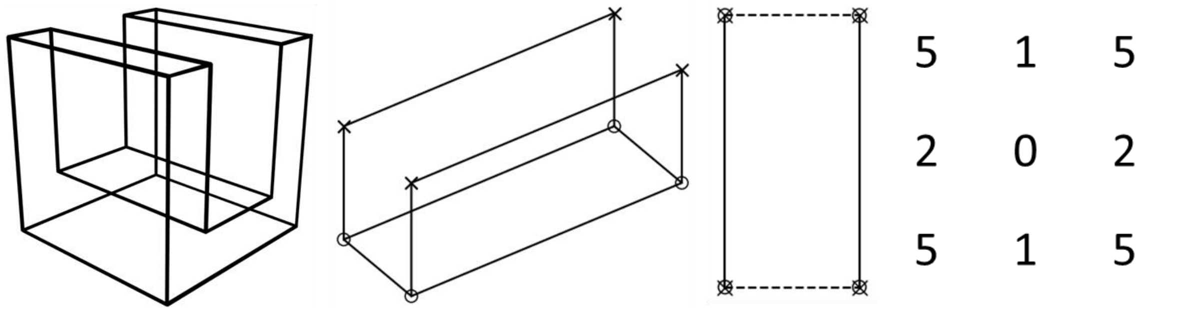


Figure 18: Example of converting a slot feature into 3×3 input matrix [150].

4.4.2.2 Graph-based representation

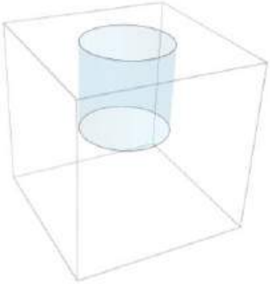
[95] proposed a method to convert the face adjacency matrix, which is generated from AAG, into a 2D-array for network input. In the input array, each element contains a vector with eight integers, which denote topology attributes such as edge type, face type, face angle type, number of loops, etc. An obvious problem of this representation is that it is verbose and has redundant data position due to the same vector definition used for representing both face characteristics and its adjacency. For example, the second number of each vector in same column is always the same, since they all indicate the face type of the same face. Another problem is that the input array is executed row by row, meanwhile the feature is defined with respect to one face and its topology relationships with a set of secondary faces. Therefore, this method neglects the topology between the secondary faces, so that the domain of recognition is limited and not able to handle interacting features. Figure 19 shows a simple example of this method.

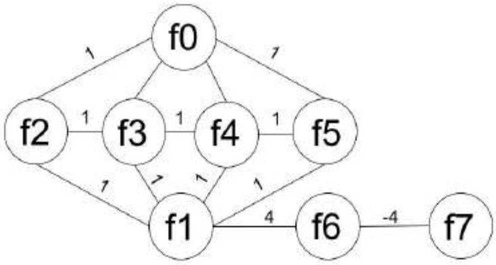
[90] developed a neat representation vector of 20 binary elements based on the AAG. In this method, the input part's AAG is first decomposed into subgraphs according to a set of heuristics. Then the adjacency matrix for each subgraph is created and refined into 12 binary numbers by interrogating a set of 12 questions about the matrix layout and the number of faces in the subgraph. The 13th to 20th elements in the representation vector is the binary form of the number of external faces linked to the subgraph. The major limitation of this method is that the design of graph decomposition heuristics limits the recognition domain to simple features with planar and simple curved faces.

In order to solve the ambiguity and incapability of handling interacting features from AAG-based methods, [22] proposed an input representation with two matrices based on the part's delta volume. One is the face adjacency matrix where the diagonal elements are assigned by numerical values between 1 and 8 representing 8 predefined face types. The off-diagonal elements are given numerical values between 0 and 9 indicating ten different connection relationships. The other matrix is named V-adjacency matrix and used for representing the virtual faces, which refer to the faces forms the delta volume but not constitute the part. The V-matrix is defined as 6×6 matrix in binary form where each row and column represent one of six directions: +x, -x, +y, -y, +z, -z. The diagonal elements indicate whether a virtual face exists (0 or 1). The other elements denote whether the corresponding pair of virtual faces connected. Figure 20 is a simple example of F-adjacency and V-adjacency matrix.

4.4.2.3 Face score vector

The concept of face score vector was originally proposed by [44] and adopted by [66],[92]. It is a compact representation in which each element corresponds to a face in the part. The element is a real number calculated by a function as $F_s=f(E_i,V_j,L_k,F_g)$, in which E_i, V_j, L_k, F_g are scores that are predefined according to the edge geometry (convex/concave edge), edge-vertex connectivity, and face geometry (planar/cylindrical face).

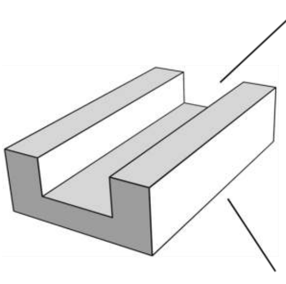




f	0	1	2	3	4	5	6	7
0	0	0	1	1	1	1	0	0
1	0	0	1	1	1	1	0	4
2	1	1	0	1	0	1	0	0
3	1	1	1	0	1	0	0	0
4	1	1	0	1	0	1	0	0
5	1	1	1	0	1	0	0	0
6	0	0	0	0	0	0	0	-4
7	0	4	0	0	0	0	-4	1

0	01111000	01011001	11011100	11011100	11011100	11011000	01001000	02000100
1	01011000	01111001	11011100	11011100	11011100	11011000	01001000	42000100
2	11011100	11011101	01111000	11011100	01010000	11011000	01001100	02001000
3	11011100	11011101	11011100	01111000	11011100	01010000	01001100	02001000
4	11011100	11011101	01010000	11011100	01111000	11011100	01001100	02001000
5	11011100	11011101	11011100	01010000	11011100	01111000	01001100	02001000
6	01011000	01011001	01011100	01011100	01011100	01011100	01011100	-42000100
7	01010100	41010101	01011000	01011000	01011000	01011000	-41000100	12101000

Figure 19: Example of transforming a simple hole feature into input vectors of ANN: (a) A hole feature, (b) AAG, (c) AAM, and (d) Input vectors [95].



$$\begin{bmatrix} 6 & 3 & 0 & 0 & 0 \\ & 6 & 3 & 0 & 0 \\ & & 6 & 0 & 0 \\ & & & 0 & 0 \\ & & & & 0 \end{bmatrix} \Rightarrow [630\ 006\ 300\ 600\ 000]$$

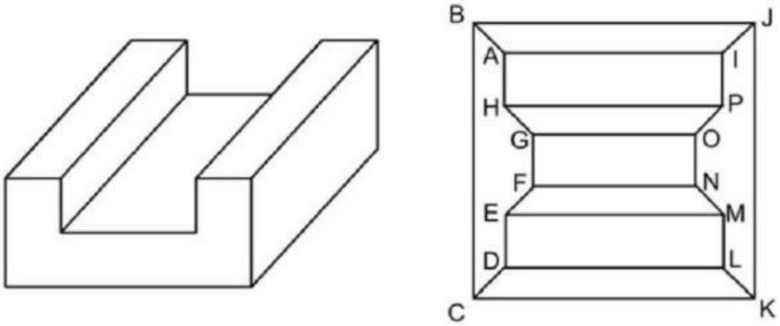
F-adjacency matrix   ANN input vector

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ & 1 & 0 & 0 & 1 & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & & & & 1 & 0 \\ & & & & & 0 \end{bmatrix} \Rightarrow [100\ 010\ 100\ 100\ 000\ 000\ 100]$$

V-adjacency matrix   ANN input vector

Figure 20: Examples of transforming slot feature into ANN input Vectors [5].

The numerical values of different edge and face geometry scores are set to reflect their geometric property of the entities. Hence, the face score vector can be considered as a measurement of the face uniqueness or face complexity based upon the convexity or concavity of surrounding region. Figure 21 shows an example of slot feature and its face score vector.



Feature type	No. of the face	Faces	Face score
SLOT	1	GFNO	(0.5+0.5+0.5+0.5)/4=0.5
	2,3	ABCDEFGH, IJKLMNOP	(6×1.5+2×0.5)/8=1.25
	4,5	GHPO, FEMN, AHPI, EMLD,	(0.5+2×1.5+0.5)/4=1.0
		ABJI, DCKL, BCKJ	(1.5+1.5+1.5+1.5)/4=1.5

Figure 21: Face score vector of a slot feature: (1.5 1.5 1.0 1.25 0.5 1.25 1.0 1.5 1.5) [5].

To better support transitional features, [81] presented a modified face score vector where the face score F_s is normalized by an equation of $(F_s + 4)/8$, since the minimal value of face score is -4 and the maximum values is +4. In order to represent machining feature families with variations in topology and geometry, [117] proposed an enhanced face score vector with 12 nodes. Nodes 1 to 7 represent face scores of the selected faces in input feature. Nodes 8, 9, 10, and 12 are used to store topologically invariant attributes within a feature family. These attributes can help to classify different feature families. Node 11 denotes the total number of faces in a candidate feature, which can aid ANN to differentiate sub-feature types in the feature families.

4.5 Hybrid Approach

Hint-based, graph-based, volume decomposition and neural networks are four basic and promising approaches in feature recognition field. They all have their advantages and limitations. Therefore, it is evident that a hybrid system adopting selective characteristics of these approaches could provide a constructive and practical solution to overcome the shortcomings of existing feature recognition systems.

For example, graph-based systems are best suited for representing isolated features and can easily incorporate new feature definitions. Nevertheless, they have difficulties in tolerating topology variations of features, so that they are incapable of handling arbitrary feature interactions. While hint-based systems are able to handle arbitrary feature interactions by taking advantage of human knowledge. However, they lack generic feature representation schemes to expand the domain of recognition to a wide variety of objects. Therefore, in order to amend the graph-based method's incapability of interacting features, [30] combined graph-based and hint-based methods. They introduced a concept of the minimal condition subgraph (MCSG) as feature hint. Six types of feature interactions and their MSCGs were defined to support graph matching. Moreover, new geometric reasoning of graph decomposition and virtual link were used to complete MSCGs and generate alternative interpretations of interacting features. In fact, this method treats feature interactions as special cases of feature and designs corresponding algorithms for them. Therefore, it still remains difficulties in extending the recognition domain to arbitrary interactions and freeform features. [100] and [101] developed a similar graph-hint hybrid system to extract subgraphs as feature hints based on the characteristics of concave features. [126] directly connected a graph-based system with a hint-based system to obtain a hybrid system, in which the isolated and interacting features are processed respectively.

Because of the strong ability of generalization and robustness of topological variations, many researches tried to integrate ANN as feature classifier with inputs generated from existing graph-based representation schemes, such as [22],[90],[95]. [71] incorporated ANN with graph and hint approaches for a novel hybrid FR system. The ANN was used as feature classifier with the inputs of vectorized feature hints, which are generated by using face loop graphs extracted from the part's EAAG. In addition, the high degree of robustness enables ANN approach to handle interacting features to some extent. [64] used the artificial neural network to convert the non-orthogonal features into a number of triangular blocks to combine into a virtual orthogonal feature, which can be recognized by FES graph-based method in the next step. Nevertheless, the considered non-orthogonal features are only composed of planar faces and be of constant depth.

Due to the remarkable exhaustive nature of volumetric representation and decomposition process, the volume decomposition methods are considered to be effective in handling the interacting features and downstream manufacturing applications. Therefore, there are various attempts to combine graph-based with volume decomposition approaches for better solutions to interacting features. One is described in [134], where the graph-based approach is used to extract and recognize features from 3D models and the volume decomposition approach is incorporated to generate delta volumes and multiple interpretations of machining operation sequences. Another type of volume-graph hybrid approach uses maximal volume decomposition to extract machining features and face adjacency graph matching to recognize features, such as [102],[109],[119]. Another example is [146], which developed a hybrid FR system by combining graph-based and convex-hull decomposition approach. The interacting features are decomposed into primitive shapes by using virtual link technique and volume manipulations. The introduction of the concept of convex-hull volume facilitates the reconstruction and recognition of decomposed features.

5 CONCLUSION

The main purpose of this review is to summarize the trends in feature recognition techniques within the past thirty years and investigate how this technology has evolved. We describe the characteristics of selected methods and their analysis in details, and for more general surveys about history of the techniques, readers are suggested to refer to [6], [38], [127]. In summary, the difficulties involved in existing feature recognition can be classified into the following categories: generic representation schemes for features should be applicable to a wide variety of objects; robust recognizers should be able to recover missing information caused by feature interactions; flexible feature domain should have learning capability to support variation and customization; and algorithmic complexity should be fairly limited so that it can providing prompt feedback to model changes. From all these drawbacks and limitations, a satisfactory system able to handle practical industrial parts still do not exist.

Most of existing feature recognition approaches were based on identification of the geometrical entities composing a feature, together with their interrelationships, against certain predefined set of rules or templates. Their target features were usually classified by some high-level characteristics such as function, usage, and manufacturing methods. In their feature representation schemes, the features are explicitly defined as a set of connected faces having certain classifiable characteristic of topology and geometry. Each feature class has a unique definition involving topological characteristics such as the number of faces, edges, vertices and their connectivity relationship, and geometric characteristic such as edge convexity, face type, normal directions and geometric constraints. Meanwhile, for feature recognition and application, there could be topology or geometry variations. These variations of topology and geometry make feature recognition methods and its library hard to be robust and expendable.

On one hand, to propose a more generalized and expendable feature recognition approach, the feature representation should go beyond individual geometric entities, and describes both the topology and geometry characteristics of a 3-D model using digital numbers, which are easy to be implemented and identifiable by classification or recognition techniques. The feature representation also shall have other advantages like expendable, unambiguous, invariance to scale, translation, and rotation. On the other hand, the conventional feature recognition methods extract a collection of surfaces or volumes from the original part as a potential feature, and then matches it with a pre-specified set of rules or templates. It is executed sequentially and logically according to the designed algorithms. To be expandable and robust, the recognition algorithms should be able to derive implicit characteristics or discover regularities from examples that are difficult to describe adequately with knowledge-based systems. The feature library can be extended to more complicated features having topology variations without extra needs to reprogram the algorithm. In addition, to handle the challenging feature interaction problem, both of the feature representation and recognition methods should demonstrate robustness to incomplete feature information. However, in most of the existing approaches, the geometric and topological variations especially the unseen ones often lead to unsuccessful recognition of the test cases.

It is clear from the research work reviewed that enormous efforts have been made in this field, however, there are few available commercial feature recognition software and little evidence of industrial implementation of the research results [6]. Moreover, feature recognition systems were developed by the encouragement from advances in computing technologies to replace expert activities in feature identification. Taking into account of the growingly complex products and manufacturing methods, the needs of developing new efficient methods of features recognition still exists.

In further perspectives, an original method development for feature recognition through implementation of artificial neural networks and other advanced artificial intelligence techniques is planned by authors. It has been known that most new designs are actually re-design activities that are based on existing conceptions but with different degrees of feature dimensional changes. It is therefore logical to conclude that feature recognition for new designs could be achieved mostly by knowledge acquisition from established database. While extraction of knowledge and pattern from designs leads to a vast body of heterogeneous, uncertain and inherently inconsistent information, which limits the development of expert systems. For artificial intelligence, on the contrary, exponential complexity, uncertainty, inconsistency, interaction of various kinds of knowledge are treated as inherent attributes of complex real-life problems since numerous methods of data representation and manipulation are designed to handle complexities.

Yang Shi, http://orcid.org/0000-0002-9579-9403
Yicha Zhang, http://orcid.org/0000-0002-5219-1748
Kaishu Xia, http://orcid.org/0000-0001-8216-6597
Ramy Harik, http://orcid.org/0000-0003-1452-9653

REFERENCES

[1] A Senthil kumar; Salim, F.; Nee, A.: Automatic recognition of design and machining features from prismatic parts, The International Journal of Advanced Manufacturing Technology, 11(2), 1996, 136-145. https://doi.org/10.1007/bf01341562

[2] Allada, V.; Anand, S.: Feature-based modelling approaches for integrated manufacturing: state-of-the-art survey and future research directions, International Journal of Computer Integrated Manufacturing, 8(6), 1995, 411-440. https://doi.org/10.1080/09511929508944670

[3] Ansaldi, S.; De Floriani, L.; Falcidieno, B.: Geometric modeling of solid objects by using a face adjacency graph representation, 19(3), 1985, 131-139.

https://doi.org/10.1145/325165.325218
[4] Ansaldi, S.; Floriani, K.; Falcidienc, B.: An Edge-Face Relational Scheme for Boundary Representations, 4(4), 1985, 319-332. https://doi.org/10.1111/j.1467-8659.1985.tb00237.x
[5] Babić, B. R.; Nešić, N.; Miljković, Z.: Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems, AI EDAM, 25(3), 2011, 289-304. https://doi.org/10.1017/s0890060410000545
[6] Babic, B.; Nesic, N.; Miljkovic, Z.: A review of automated feature recognition with rule-based pattern recognition, Computers in Industry, 59(4), 2008, 321-337. https://doi.org/10.1016/j.compind.2007.09.001
[7] Bhandarkar, M. P.; Nagi, R.: STEP-based feature extraction from STEP geometry for agile manufacturing, Computers in Industry, 41(1), 2000, 3-24. https://doi.org/10.1016/s0166-3615(99)00040-8
[8] Brousseau, Emmanuel Bruno Jean Paul: Intelligent Techniques for Automatic Feature Recognition in CAD Models, Doctoral, University of Wales, Cardiff, 2004.
[9] Brousseau, E.; Dimov, S.; Setchi, R.: Knowledge acquisition techniques for feature recognition in CAD models, Journal of Intelligent Manufacturing, 19(1), 2008, 21-32. https://doi.org/10.1007/s10845-007-0043-7
[10] Chan, A.; Case, K.: Process planning by recognizing and learning machining features, International Journal of Computer Integrated Manufacturing, 7(2), 1994, 77-99. https://doi.org/10.1080/09511929408944597
[11] Chazelle, B. M.: Convex decompositions of polyhedra, 1981, 70-79. https://doi.org/10.1145/800076.802459
[12] Chen, Y.; Lee, H.: A neural network system feature recognition for two-dimensional, International Journal of Computer Integrated Manufacturing, 11(2), 1998, 111-117. https://doi.org/10.1080/095119298130859
[13] Choi, B.; Barash, M.; Anderson, D.: Automatic recognition of machined surfaces from a 3D solid model, Computer-Aided Design, 16(2), 1984, 81-86. https://doi.org/10.1016/0010-4485(84)90164-7
[14] Chuang, J.; Wang, P.; Wu, M.: Automatic classification of block-shaped parts based on their 2D projections, Computers & Industrial Engineering, 36(3), 1999, 697-718. https://doi.org/10.1016/s0360-8352(99)00160-6
[15] Chuang, S.; Henderson, M. R.: Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs, Computer-Aided Design, 22(6), 1990, 377-387. https://doi.org/10.1016/0010-4485(90)90088-t
[16] Coles, J. K.; Crawford, R. H.; Wood, K. L.: Form feature recognition using base volume decomposition, Paper presented at Proceedings of the 1994 ASME Design Technical Conferences. Part 1 (of 2), Minneapolis, MN, USA, 9/11/94 - 9/14/94, 1994, 281-297.
[17] Corney, J.; Clark, D. E.: Method for finding holes and pockets that connect multiple faces in 2 1/2D objects, Computer-Aided Design, 23(10), 1991, 658-668. https://doi.org/10.1016/0010-4485(91)90018-r
[18] De Floriani, L.: Feature extraction from boundary models of three-dimensional objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(8), 1989, 785-798. https://doi.org/10.1109/34.31442
[19] De Floriani, L.; Falcidieno, B.: A hierarchical boundary model for solid object representation, ACM Transactions on Graphics (TOG), 7(1), 1988, 42-60. https://doi.org/10.1145/42188.46164
[20] Dimov, S.; Brousseau, E.; Setchi, R.: A hybrid method for feature recognition in computer-aided design models, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 221(1), 2007, 79-96. https://doi.org/10.1243/09544054JEM437
[21] Ding, L.; Matthews, J.: A contemporary study into the application of neural network techniques employed to automate CAD/CAM integration for die manufacture, Computers & Industrial Engineering, 57(4), 2009, 1457-1471. https://doi.org/10.1016/j.cie.2009.01.006

[22] Ding, L.; Yue, Y.: Novel ANN-based feature recognition incorporating design by features, Computers in Industry, 55(2), 2004, 197-222. https://doi.org/10.1016/j.compind.2004.02.002
[23] DONALDSON, I. A.; CORNEY, J. R.: Rule-based feature recognition for 2· 5D machined components, International Journal of Computer Integrated Manufacturing, 6(1-2), 1993, 51-64. https://doi.org/10.1080/09511929308944555
[24] Dong, J.; Vijayan, S.: Features extraction with the consideration of manufacturing processes, International Journal of Production Research, 35(8), 1997, 2135-2156. https://doi.org/10.1080/002075497194778
[25] Falcidieno, B.; Giannini, F.: Extraction and organization of form features into a structured boundary model, 87, 1987, 349-359. http://dx.doi.org/10.2312/egtp.19871019
[26] Ferreira, J.; Hinduja, S.: Convex hull-based feature-recognition method for 2.5 D components, Computer-Aided Design, 22(1), 1990, 41-49. https://doi.org/10.1016/0010-4485(90)90028-b
[27] Fields, M. C.; Anderson, D. C.: Fast feature extraction for machining applications, Computer-Aided Design, 26(11), 1994, 803-813. https://doi.org/10.1016/0010-4485(94)90094-9
[28] Fu, M.; Ong, S.; Lu, W. F.; Lee, I.; Nee, A. Y.: An approach to identify design and manufacturing features from a data exchanged part model, Computer-Aided Design, 35(11), 2003, 979-993. https://doi.org/10.1016/s0010-4485(02)00160-4
[29] Gao, J.; Zheng, D.; Gindy, N.: Extraction of machining features for CAD/CAM integration, The International Journal of Advanced Manufacturing Technology, 24(7-8), 2004, 573-581. https://doi.org/10.1007/s00170-003-1882-9
[30] Gao, S.; Shah, J. J.: Automatic recognition of interacting machining features based on minimal condition subgraph, Computer-Aided Design, 30(9), 1998, 727-739. https://doi.org/10.1016/s0010-4485(98)00033-5
[31] Gao, S.; Zhao, W.; Lin, H.; Yang, F.; Chen, X.: Feature suppression based CAD mesh model simplification, Computer-Aided Design, 42(12), 2010, 1178-1188. https://doi.org/10.1016/j.cad.2010.05.010
[32] Gavankar, P.; Henderson, M. R.: Graph-based extraction of protrusions and depressions from boundary representations, Computer-Aided Design, 22(7), 1990, 442-450. https://doi.org/10.1016/0010-4485(90)90109-p
[33] General Dynamics Corporation: Volume decomposition algorithm - Final report, R-82-ANC-01, CAM-I, Inc, Arlington, Texas, 1985.
[34] Geng, W.; Chen, Z.; He, K.; Wu, Y.: Feature recognition and volume generation of uncut regions for electrical discharge machining, Advances in Engineering Software, 91, 2016, 51-62. https://doi.org/10.1016/j.advengsoft.2015.10.005
[35] Gindy, N.: A hierarchical structure for form features, The International Journal of Production Research, 27(12), 1989, 2089-2103. https://doi.org/10.1080/00207548908942677
[36] Gupta, S. K.: No title, Automated manufacturability analysis of machined parts, 1995, http://hdl.handle.net/1903/5693.
[37] Han, J.; Kang, M.; Choi, H.: STEP-based feature recognition for manufacturing cost optimization, Computer-Aided Design, 33(9), 2001, 671-686. https://doi.org/10.1016/s0010-4485(01)00071-9
[38] Han, J.; Pratt, M.; Regli, W. C.: Manufacturing feature recognition from solid models: a status report, IEEE Transactions on Robotics and Automation, 16(6), 2000, 782-796. https://doi.org/10.1109/70.897789
[39] Han, J.; Requicha, A. A.: Feature recognition from CAD models, IEEE Computer Graphics and Applications, 18(2), 1998, 80-94. https://doi.org/10.1109/38.656791
[40] Han, J.; Requicha, A. A.: Integration of feature based design and feature recognition, Computer-Aided Design, 29(5), 1997, 393-403. https://doi.org/10.1016/s0010-4485(96)00079-6
[41] Hofmann, C. M.: Geometric and solid modeling: an introduction, Morgan Kaufmann, 1989. https://doi.org/10.1137/1034067

[42] Huang, Z.; Yip-Hoi, D.: High-level feature recognition using feature relationship graphs, Computer-Aided Design, 34(8), 2002, 561-582. https://doi.org/10.1016/s0010-4485(01)00128-2
[43] Hummel, K.: Coupling rule-based and object-oriented programming for the classification of machined features, 1989. https://www.osti.gov/biblio/5654389
[44] Hwang, J.: Applying the perceptron to three-dimensional feature recognition, 1992.
[45] International Organisation for Standardisation: Industrial automation systems and integration -- Product data representation and exchange -- Part 224: Application protocol: Mechanical product definition for process planning using machining features, ISO 10303-224:2006(E).
[46] Ji, Q.; Marefat, M. M.: Bayesian approach for extracting and identifying features, Computer-Aided Design, 27(6), 1995, 435-454. https://doi.org/10.1016/0010-4485(95)00017-l
[47] Ji, Q.; Marefat, M. M.: A Dempster–Shafer approach for recognizing machine features from CAD models, Pattern Recognition, 36(6), 2003, 1355-1368. https://doi.org/10.1016/s0031-3203(02)00260-1
[48] Jiao, X.; Heath, M. T.: Feature detection for surface meshes, 2002, 705-714.
[49] Jones, T.; Reidsema, C.; Smith, A.: Automated Feature Recognition System for supporting conceptual engineering design, International Journal of Knowledge-Based and Intelligent Engineering Systems, 10(6), 2006, 477-492. https://doi.org/10.3233/kes-2006-10606
[50] Joshi, S.; Chang, T.: Feature extraction and feature based design approaches in the development of design interface for process planning, Journal of Intelligent Manufacturing, 1(1), 1990, 1-15. https://doi.org/10.1007/bf01471338
[51] Joshi, S.; Chang, T.: Graph-based heuristics for recognition of machined features from a 3D solid model, Computer-Aided Design, 20(2), 1988, 58-66. https://doi.org/10.1016/0010-4485(88)90050-4
[52] Jun, Y. T.; Raja, V. H.: Extracting geometric attributes directly from scanned data sets for feature recognition, International Journal of Computer Integrated Manufacturing, 15(1), 2002, 50-61. https://doi.org/10.1080/09511920110035021
[53] Kailash, S.; Zhang, Y.; Fuh, J. Y.: A volume decomposition approach to machining feature extraction of casting and forging components, Computer-Aided Design, 33(8), 2001, 605-617. https://doi.org/10.1016/s0010-4485(00)00107-x
[54] Kamrani, A. K.; Nasr, E. A.: Engineering design and rapid prototyping, Springer Science & Business Media, 2010. https://doi.org/10.1007/978-0-387-95863-7
[55] Kim, B. C.; Mun, D.: Enhanced volume decomposition minimizing overlapping volumes for the recognition of design features, Journal of Mechanical Science and Technology, 29(12), 2015, 5289-5298. https://doi.org/10.1007/s12206-015-1131-9
[56] Kim, B. C.; Mun, D.: Feature-based simplification of boundary representation models using sequential iterative volume decomposition, Computers & Graphics, 38, 2014, 97-107. https://doi.org/10.1016/j.cag.2013.10.031
[57] Kim, Y. S.: Recognition of form features using convex decomposition, Computer-Aided Design, 24(9), 1992, 461-476. https://doi.org/10.1016/0010-4485(92)90027-8
[58] Kim, Y. S.; Wang, E.; Rho, H. M.: Geometry-based machining precedence reasoning for feature-based process planning, International Journal of Production Research, 39(10), 2001, 2077-2103. https://doi.org/10.1080/00207540110038513
[59] Kim, Y. S.; Wilde, D.: A convergent convex decomposition of polyhedral objects, Journal of Mechanical Design, 114(3), 1992, 468-476. https://doi.org/10.1115/1.2926575
[60] Kim, Y. S.; Wilde, D.: A convex decomposition using convex hulls and local cause of its non-convergence, Journal of Mechanical Design, 114(3), 1992, 459-467. https://doi.org/10.1115/1.2926574
[61] Koo, S.; Lee, K.: Wrap-around operation to make multi-resolution model of part and assembly, Computers & Graphics, 26(5), 2002, 687-700. https://doi.org/10.1016/s0097-8493(02)00124-3
[62] Kyprianou, L. K.: Shape classification in computer-aided design, Doctoral dissertation, University of Cambridge, 1980.

[63] Laakko, T.; Mäntylä, M.: Feature modelling by incremental feature recognition, Computer-Aided Design, 25(8), 1993, 479-492. https://doi.org/10.1016/0010-4485(93)90079-4

[64] Lam, S.; Wong, T.: Recognition of machining features-a hybrid approach, International Journal of Production Research, 38(17), 2000, 4301-4316. https://doi.org/10.1080/00207540050205109

[65] Langerak, T. R.; Vergeest, J. S.: Feature recognition of user-defined freeform features, Computer-Aided Design and Applications, 4(1-4), 2007, 529-538. https://doi.org/10.1080/16864360.2007.10738572

[66] Lankalapalli, K.; Chatterjee, S.; Chang, T.: Feature recognition using ART2: a self-organizing neural network, Journal of Intelligent Manufacturing, 8(3), 1997, 203-214. https://doi.org/10.1023/A:1018521207901

[67] Lee, J. Y.; Kim, K.: A feature-based approach to extracting machining features, Computer-Aided Design, 30(13), 1998, 1019-1035. https://doi.org/10.1016/s0010-4485(98)00055-4

[68] Lee, Y.; Fu, K.: Machine understanding of CSG: extraction and unification of manufacturing features, IEEE Computer Graphics and Applications, 7(1), 1987, 20-32. https://doi.org/10.1109/mcg.1987.277024

[69] Lentz, D. H.; Sowerby, R.: Feature extraction of concave and convex regions and their intersections, Computer-Aided Design, 25(7), 1993, 421-437. https://doi.org/10.1016/0010-4485(93)90004-8

[70] Li, B.; Liu, J.: Detail feature recognition and decomposition in solid model, Computer-Aided Design, 34(5), 2002, 405-414. https://doi.org/10.1016/s0010-4485(01)00118-x

[71] Li, W.; Ong, S.; Nee, A.: A hybrid method for recognizing interacting machining features, International Journal of Production Research, 41(9), 2003, 1887-1908. https://doi.org/10.1080/0020754031000123868

[72] Li, W.; Ong, S.; Nee, A.: Recognition of overlapping machining features based on hybrid artificial intelligent techniques, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 214(8), 2000, 739-744. https://doi.org/10.1243/0954405001518107

[73] Li, Y.; Ding, Y.; Mou, W.; 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. https://doi.org/10.1243/09544054jem1634

[74] Lin, A. C.; Lin, S.: A volume decomposition approach to process planning for prismatic parts with depression and protrusion design features, International Journal of Computer Integrated Manufacturing, 11(6), 1998, 548-563. https://doi.org/10.1080/095119298130570

[75] Liu, C.; Perng, D.; Chen, Z.: Automatic form feature recognition and 3D part reconstruction from 2D CAD data, Computers & Industrial Engineering, 26(4), 1994, 689-707. https://doi.org/10.1016/0360-8352(94)90006-x

[76] Lockett, H. L.; Guenov, M. D.: Graph-based feature recognition for injection moulding based on a mid-surface approach, Computer-Aided Design, 37(2), 2005, 251-262. https://doi.org/10.1016/j.cad.2004.06.010

[77] 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. https://doi.org/10.1016/s0010-4485(00)00122-6

[78] Marchetta, M. G.; Forradellas, R. Q.: An artificial intelligence planning approach to manufacturing feature recognition, Computer-Aided Design, 42(3), 2010, 248-256. https://doi.org/10.1016/j.cad.2009.11.007

[79] Marefat, M.; Kashyap, R. L.: Automatic construction of process plans from solid model representations, IEEE transactions on systems, man, and cybernetics, 22(5), 1992, 1097-1115. https://doi.org/10.1109/21.179847

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

[81] Marquez, M.; Gill, R.; White, A.: Application of neural networks in feature recognition of mould reinforced plastic parts, Concurrent Engineering, 7(2), 1999, 115-122. https://doi.org/10.1177/1063293x9900700204
[82] Martin, P.: Some aspects of integrated production and manufacturing. In: Bramley, A., Brissaud, D., Coutellier, D., McMahon, C. eds., Advances in Integrated Design and Manufacturing in Mechanical Engineering , Springer, Amsterdam, 2005, 215-226. https://doi.org/10.1007/1-4020-3482-2_17
[83] Martino, S.; Kim, Y.: Cylindrical features in form feature recognition using convex decomposition, Proc. IFIP Conf. on Feature Modeling and Recognition in Advanced CAD/CAM Systems, 1994.
[84] Meeran, S.; Taib, J.; Afzal, M.: Recognizing features from engineering drawings without using hidden lines: a framework to link feature recognition and inspection systems, International Journal of Production Research, 41(3), 2003, 465-495. https://doi.org/10.1080/00207540210148871
[85] Meeran, S.; Pratt, M. J.: Automated feature recognition from 2D drawings, Computer-Aided Design, 25(1), 1993, 7-17. https://doi.org/10.1016/0010-4485(93)90061-r
[86] Meeran, S.; Zulkifli, A.: Recognition of simple and complex interacting non-orthogonal features, Pattern Recognition, 35(11), 2002, 2341-2353. https://doi.org/10.1016/s0031-3203(01)00223-0
[87] Miao, H. K.; Sridharan, N.; Shah, J. J.: CAD-CAM integration using machining features, International Journal of Computer Integrated Manufacturing, 15(4), 2002, 296-318. https://doi.org/10.1080/09511920110077502
[88] Narang, R. V.: An application-independent methodology of feature recognition with explicit representation of feature interaction, Journal of Intelligent Manufacturing, 7(6), 1996, 479-486. https://doi.org/10.1007/bf00122837
[89] Nasr, E. S. A.; Kamrani, A. K.: A new methodology for extracting manufacturing features from CAD system, Computers & Industrial Engineering, 51(3), 2006, 389-415. https://doi.org/10.1016/j.cie.2006.08.004
[90] Nezis, K.; Vosniakos, G.: Recognizing 212D shape features using a neural network and heuristics, Computer-Aided Design, 29(7), 1997, 523-539. https://doi.org/10.1016/s0010-4485(97)00003-1
[91] Öztürk, N.; Öztürk, F.: Hybrid neural network and genetic algorithm based machining feature recognition, Journal of Intelligent Manufacturing, 15(3), 2004, 287-298. https://doi.org/10.1023/b:jims.0000026567.63397.d5
[92] Öztürk, N.; Öztürk, F.: Neural network based non-standard feature recognition to integrate CAD and CAM, Computers in Industry, 45(2), 2001, 123-135. https://doi.org/10.1016/s0166-3615(01)00090-2
[93] Parienté, F.; Kim, Y. S.: Incremental and localized update of convex decomposition used for form feature recognition, Computer-Aided Design, 28(8), 1996, 589-602. https://doi.org/10.1016/0010-4485(95)00074-7
[94] Peters, T. J.: Encoding mechanical design features for recognition via neural nets, Research in Engineering Design, 4(2), 1992, 67-74. https://doi.org/10.1007/bf01580145
[95] Prabhakar, S.; Henderson, M. R.: Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models, Computer-Aided Design, 24(7), 1992, 381-393. https://doi.org/10.1016/0010-4485(92)90064-h
[96] Pratt, M. J.: Applications of feature recognition in the product life-cycle, International Journal of Computer Integrated Manufacturing, 6(1-2), 1993, 13-19. https://doi.org/10.1080/09511929308944550
[97] Pratt, M. J.: A hybrid feature based modelling system. In: Krause, F. L., Jansen, H. eds., Advanced Geometric Modelling for Engineering Applications, Elsevier Science Publishers, North-Holland, Amsterdam, 1990, 177-189.
[98] Pratt, M. J.: Aspects of form feature modelling. In: Anonymous Geometric Modeling, Springer, 1991, 227-250. https://doi.org/10.1007/978-3-642-76404-2_10

[99] Qamhiyah, A. Z.; Venter, R.; Benhabib, B.: Geometric reasoning for the extraction of form features, Computer-Aided Design, 28(11), 1996, 887-903. https://doi.org/10.1016/0010-4485(96)00015-2
[100] Rahmani, K.; Arezoo, B.: Boundary analysis and geometric completion for recognition of interacting machining features, Computer-Aided Design, 38(8), 2006, 845-856. https://doi.org/10.1016/j.cad.2006.04.015
[101] Rahmani, K.; Arezoo, B.: A hybrid hint-based and graph-based framework for recognition of interacting milling features, Computers in Industry, 58(4), 2007, 304-312. https://doi.org/10.1016/j.compind.2006.07.001
[102] Rameshbabu, V.; Shunmugam, M.: Hybrid feature recognition method for setup planning from STEP AP-203, Robotics and Computer-Integrated Manufacturing, 25(2), 2009, 393-408. https://doi.org/10.1016/j.rcim.2007.09.014
[103] Ranjan, R.; Kumar, N.; Pandey, R. K.; Tiwari, M.: Automatic recognition of machining features from a solid model using the 2D feature pattern, The International Journal of Advanced Manufacturing Technology, 26(7-8), 2005, 861-869. https://doi.org/10.1007/s00170-003-2059-2
[104] Regli, W. C.; Gupta, S. K.; Nau, D. S.: Extracting alternative machining features: An algorithmic approach, Research in Engineering Design, 7(3), 1995, 173-192. https://doi.org/10.1007/bf01638098
[105] Regli, W. C.; Gupta, S. K.; Nau, D. S.: Feature recognition for manufacturability analysis, 1994, http://hdl.handle.net/1903/5490.
[106] Sakurai, H.; Gossard, D. C.: Shape feature recognition from 3D solid models, ASME Computers in Engineering, San Francisco, 1988. https://doi.org/10.1109/38.59033
[107] Sakurai, H.: Volume decomposition and feature recognition: Part 1—polyhedral objects, Computer-Aided Design, 27(11), 1995, 833-843. https://doi.org/10.1016/0010-4485(95)00007-0
[108] Sakurai, H.; Chin, C.: Definition and recognition of volume features for process planning. In: Anonymous Manufacturing Research and Technology, Elsevier, 1994, 65-80. https://doi.org/10.1016/b978-0-444-81600-9.50009-2
[109] Sakurai, H.; Dave, P.: Volume decomposition and feature recognition, Part II: curved objects, Computer-Aided Design, 28(6-7), 1996, 519-537. https://doi.org/10.1016/0010-4485(95)00067-4
[110] Sakurai, H.; Gossard, D. C.: Recognizing shape features in solid models, IEEE Computer Graphics and Applications, 10(5), 1990, 22-32. https://doi.org/10.1109/38.59033
[111] 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. https://doi.org/10.1115/1.1345522
[112] Shah, J. J.; Mäntylä, M.: Parametric and feature-based CAD/CAM: concepts, techniques, and applications, John Wiley & Sons, New York, 1995. https://doi.org/10.5860/choice.33-6352
[113] Shah, J. J.; Shen, Y.; Shirur, A.: Determination of machining volumes from extensible sets of design features. In: Anonymous Manufacturing Research and Technology, Elsevier, 1994, 129-157. https://doi.org/10.1016/b978-0-444-81600-9.50012-2
[114] Sommerville, M.; Clark, D. E.; Corney, J. R.: Viewer-centered geometric feature recognition, Journal of Intelligent Manufacturing, 12(4), 2001, 359-375. https://doi.org/10.1023/A:1011219517642
[115] Subrahmanyam, S.; Wozny, M.: An overview of automatic feature recognition techniques for computer-aided process planning, Computers in Industry, 26(1), 1995, 1-21. https://doi.org/10.1016/0166-3615(95)80003-4
[116] Sunil, V.; Pande, S.: Automatic recognition of features from freeform surface CAD models, Computer-Aided Design, 40(4), 2008, 502-517. https://doi.org/10.1016/j.cad.2008.01.006
[117] Sunil, V.; Pande, S.: Automatic recognition of machining features using artificial neural networks, The International Journal of Advanced Manufacturing Technology, 41(9-10), 2009, 932-947. https://doi.org/10.1007/s00170-008-1536-z

[118] Tang, K.; Woo, T.: Algorithmic aspects of alternating sum of volumes. Part 1: Data structure and difference operation, Computer-Aided Design, 23(5), 1991, 357-366. https://doi.org/10.1016/0010-4485(91)90029-v
[119] Tseng, Y.; Joshi, S. B.: Recognizing multiple interpretations of interacting machining features, Computer-Aided Design, 26(9), 1994, 667-688. https://doi.org/10.1016/0010-4485(94)90018-3
[120] Tyan, L.; Devarajan, V.: Automatic identification of non-intersecting machining features from 2D CAD input, Computer-Aided Design, 30(5), 1998, 357-366. https://doi.org/10.1016/s0010-4485(97)00084-5
[121] Vandenbrande, J. H.; Requicha, A. A.: Spatial reasoning for the automatic recognition of machinable features in solid models, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(12), 1993, 1269-1285. https://doi.org/10.1109/34.250845
[122] Venuvinod, P. K.; Wong, S.: A graph-based expert system approach to geometric feature recognition, Journal of Intelligent Manufacturing, 6(3), 1995, 155-162. https://doi.org/10.1007/bf00171444
[123] Venuvinod, P.; Yuen, C.; Merchant, M.: Efficient automated geometric feature recognition through feature coding, CIRP Annals-Manufacturing Technology, 43(1), 1994, 413-416. https://doi.org/10.1016/s0007-8506(07)62242-2
[124] Verma, A.; Rajotia, S.: Feature vector: a graph-based feature recognition methodology, International Journal of Production Research, 42(16), 2004, 3219-3234. https://doi.org/10.1080/00207540410001699408
[125] Verma, A.; Rajotia, S.: A hint-based machining feature recognition system for 2.5 D parts, International Journal of Production Research, 46(6), 2008, 1515-1537. https://doi.org/10.1080/00207540600919373
[126] Verma, A. K.; Rajotia, S.: A hybrid machining Feature Recognition system, International Journal of Manufacturing Research, 4(3), 2009, 343-361. https://doi.org/10.1504/ijmr.2009.026578
[127] Verma, A. K.; Rajotia, S.: A review of machining feature recognition methodologies, International Journal of Computer Integrated Manufacturing, 23(4), 2010, 353-368. https://doi.org/10.1080/09511921003642121
[128] Vosniakos, G.; Davies, B.: A shape feature recognition framework and its application to holes in prismatic parts, The International Journal of Advanced Manufacturing Technology, 8(6), 1993, 345-351. https://doi.org/10.1007/bf01751095
[129] Waco, D. L.; Kim, Y. S.: Geometric reasoning for machining features using convex decomposition, Computer-Aided Design, 26(6), 1994, 477-489. https://doi.org/10.1016/0010-4485(94)90069-8
[130] Wang, J.; Liu, S.: Hopfield neural network-based automatic recognition for 3-d features, 3, 1993, 2121-2124. https://doi.org/10.1109/ijcnn.1993.714143
[131] Wang, M.; Chamberlain, M. A.; Joneja, A.; Chang, T.: Manufacturing feature extraction and machined volume decomposition in a computer-integrated feature-based design and manufacturing planning environment, Computers in Industry, 23(1-2), 1993, 75-86. https://doi.org/10.1016/0166-3615(93)90117-j
[132] Weiler, K. J.: Topological Structures for Geometric Modeling, PhD Thesis, Rensselaer Polytechnic Institute, 1986.
[133] Wilson, P.; Pratt, M.: Requirements for support of form features in a solid modelling system, Tech. Report, Geometric Modelling Project, 1985.
[134] Wong, T.; Lam, S.: Automatic recognition of machining features from computer aided design part models, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 214(6), 2000, 515-520. https://doi.org/10.1243/0954405001517810
[135] Woo, T. C.: Feature extraction by volume decomposition, 1982, 76-94.
[136] Woo, Y.: Fast cell-based decomposition and applications to solid modeling, Computer-Aided Design, 35(11), 2003, 969-977. https://doi.org/10.1016/s0010-4485(02)00144-6

[137] Woo, Y.; Lee, S. H.: Volumetric modification of solid CAD models independent of design features, Advances in Engineering Software, 37(12), 2006, 826-835. https://doi.org/10.1016/j.advengsoft.2006.04.002
[138] Woo, Y.; Sakurai, H.: Recognition of maximal features by volume decomposition, Computer-Aided Design, 34(3), 2002, 195-207. https://doi.org/10.1016/s0010-4485(01)00080-x
[139] Wu, M.; Jen, S.: A neural network approach to the classification of 3D prismatic parts, The International Journal of Advanced Manufacturing Technology, 11(5), 1996, 325-335. https://doi.org/10.1007/bf01845691
[140] Wu, M.; Lit, C.: Analysis on machined feature recognition techniques based on B-rep, Computer-Aided Design, 28(8), 1996, 603-616. https://doi.org/10.1016/0010-4485(95)00075-5
[141] Xu, X.; Hinduja, S.: Recognition of rough machining features in 212D components, Computer-Aided Design, 30(7), 1998, 503-516. https://doi.org/10.1016/s0010-4485(97)00090-0
[142] Xu, X.: Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control, Information Science Reference, 2009. https://doi.org/10.4018/978-1-59904-714-0
[143] Yang, M.; Marefat, M. M.: Constraint-based feature recognition: Handling non-uniqueness in feature interactions, 2, 1996, 1505-1510. https://doi.org/10.1109/robot.1996.506918
[144] Yildiz, A.; Öztürk, N.; Kaya, N.; Öztürk, F.: Integrated optimal topology design and shape optimization using neural networks, Structural and Multidisciplinary Optimization, 25(4), 2003, 251-260. https://doi.org/10.1007/s00158-003-0300-0
[145] Yuen, C.; Venuvinod, P.: Geometric feature recognition: coping with the complexity and infinite variety of features, International Journal of Computer Integrated Manufacturing, 12(5), 1999, 439-452. https://doi.org/10.1080/095119299130173
[146] Zhang, C.; Chan, K.; Chen, Y.: A hybrid method for recognizing feature interactions, Integrated Manufacturing Systems, 9(2), 1998, 120-128. https://doi.org/10.1108/09576069810202078
[147] Zhang, X.; Nassehi, A.; Newman, S. T.: Feature recognition from CNC part programs for milling operations, The International Journal of Advanced Manufacturing Technology, 70(1-4), 2014, 397-412. https://doi.org/10.1007/s00170-013-5275-4
[148] Zhu, H.; Menq, C.: B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition, Computer-Aided Design, 34(2), 2002, 109-123. https://doi.org/10.1016/s0010-4485(01)00056-2
[149] Zulkifli, A.; Meeran, S.: Decomposition of interacting features using a Kohonen self-organizing feature map neural network, Engineering Applications of Artificial Intelligence, 12(1), 1999, 59-78. https://doi.org/10.1016/s0952-1976(98)00050-5
[150] Zulkifli, A.; Meeran, S.: Feature patterns in recognizing non-interacting and interacting primitive, circular and slanting features using a neural network, International Journal of Production Research, 37(13), 1999, 3063-3100. https://doi.org/10.1080/002075499190428