

# Chapter 6

## Intelligent Design and Manufacturing

Emad S. Abouel Nasr<sup>1</sup> and Ali K. Kamrani<sup>2</sup>

**Abstract** The progressive opportunity of international markets has led to significant new competitive pressures on industry. Recently, this has seen changes in organizational structures at the product design level through the prologue of computer-integrated manufacturing (CIM) and concurrent engineering (CE) philosophy, and is now seeing changes in industry structures as companies build worldwide manufacturing relationships. Automatic feature recognition from computer-aided design (CAD) systems plays an important key toward CAD/computer-aided manufacturing (CAM) integration. Different CAD packages store the information related to the design in their own databases. Structures of these databases are different from each other. As a result, no common or standard structure that can be used by all CAD packages has been developed so far. For that reason, this chapter will propose an intelligent feature recognition methodology to develop a feature recognition system which has the ability to communicate with various CAD/CAM systems. The system takes a neutral file in initial graphics exchange specification (IGES) format for 3-D prismatic parts as input and translates the information in the file to manufacturing information. The boundary representation (B-rep) geometrical information of the part is analyzed by a feature recognition program on the basis of object-oriented and geometric reasoning approaches. A feature recognition algorithm is used to recognize different features such as step and holes.

### 6.1 Introduction

The main objective of any manufacturing organization is to produce high-quality products at the lowest possible cost. The growing complexity of achieving this objective, with sharply rising costs and increased competition, has forced the industry to look for alternatives to the traditional approaches to design, manufacturing, and management. Many industries are adopting a concurrent engineering (CE) approach to develop and

---

<sup>1</sup> Assistant Professor, Mechanical Engineering Department, Faculty of Engineering at Helwan, Helwan University, Helwan, Cairo, Egypt

<sup>2</sup> Associate Professor, Department of Industrial Engineering, University of Houston, Houston, Texas 77004

produce new products in the most efficient manner. Computer-aided process planning (CAPP) systems can help reduce the planning time and increase consistency and efficiency [20]. However, the main problem of transferring the computer-aided design (CAD) data to a downstream computer-aided manufacturing (CAM) system, in order to develop computer-integrated manufacturing (CIM) environment, is the lack of neutral formats as well as content to convey the CAD information [27, 39].

Recently, this has seen changes in organizational structures at the product design level through the prologue of CIM and CE philosophy, and is now seeing changes in industry structures as companies build worldwide manufacturing relationships [36, 38]. Taking these issues into considerations leads to the recognition that the integration between design and manufacturing needs to be made to ensure business competitiveness. In order to achieve the integration of design and manufacturing, understanding of how the manufacturing information can be obtained directly from the CAD system must be addressed [13, 29, 31].

CAD and CAM systems are based on modeling geometric data. The usefulness of CAD/CAM systems is the ability to visualize product design, support design analysis, and link to the generation of part programmers for manufacturing [32]. However, CAD/CAM systems need the standardization that makes them have the ability to communicate to each other. Different CAD or geometric modeling packages store the information related to the design in their own databases and the structures of these databases are different from each other. As a result, no common or standard structure has so far been developed yet that can be used by all CAD packages. On the contrary, the conventional approach to feature extraction is accomplished by the human planner examining the part and recognizing the features designed into the part. Automated feature recognition can best be facilitated by CAD systems capable of generating the product geometry based on features, thereby making it possible to capture information about tolerance, surface finish, etc. [12]. However, such CAD systems are not yet mature and their wide usage in different application domains remains to be seen. For that reason, in this chapter, a methodology for feature analysis and extraction of prismatic parts for CAM applications is developed and presented. This approach aims to achieve the integration between CAD and CAM.

Including this introductory section, the chapter is organized into six sections. The problem statement is addressed in Sect. 6.2. Section 6.3 describes the literature review of the previous research efforts in the area of feature extraction and recognition. The proposed approach for extraction of manufacturing entities from initial graphics exchange specification (IGES) file as a standard format is presented in Sect. 6.4. The implementation of the suggested approach is demonstrated through an example in Sect. 6.5. Finally, Sect. 6.6 presents conclusions.

## 6.2 Problem Statement

Different CAD or geometric modeling packages store the information related to the design in their own databases. Structures of these databases are different from each other. As a result, no common or standard structure that can be used by all CAD

packages has been developed so far. For that reason, this research will try to develop an intelligent feature recognition methodology which has the ability to communicate with the different CAD/CAM systems by using object-oriented and geometric reasoning approaches.

### 6.3 Literature Review

The recognized features and their relationships due to the feature recognition process are used to restructure the part. A feature has a higher level of abstraction than a geometric primitive in the traditional solid model [8]. Features not only represent the shape but should also contain information on its functions and interrelationship with other features [11].

The most widely used representations methods are boundary representation (B-rep) and constructive solid geometry (CSG). The CSG of the solid model is specified with a set of Boolean operations and a set of 3-D primitive solids as shown in Fig. 6.1. On the contrary, B-rep is one of the solid modeling methods that

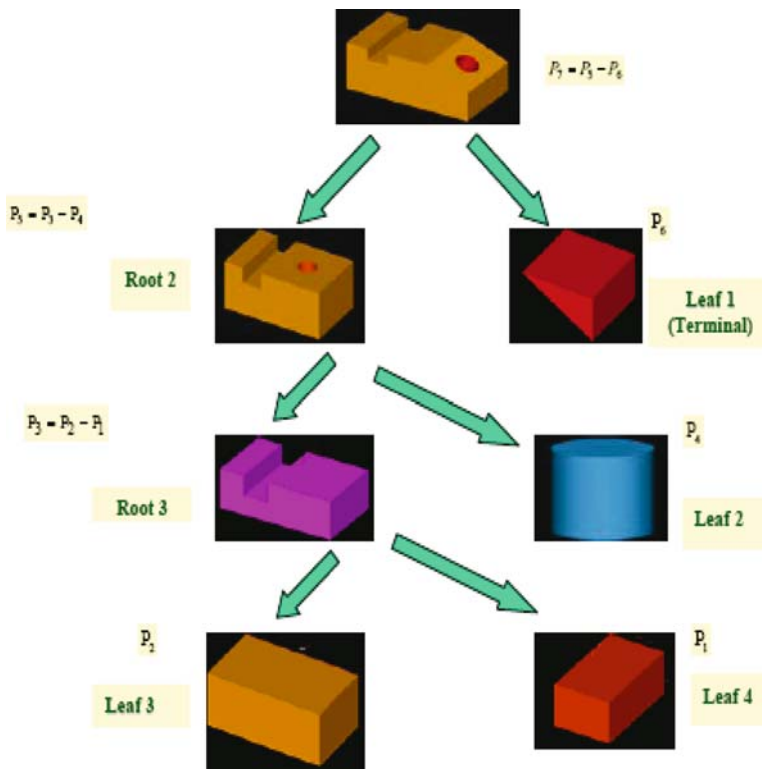


Fig. 6.1 Constructive solid geometry (CSG) representation

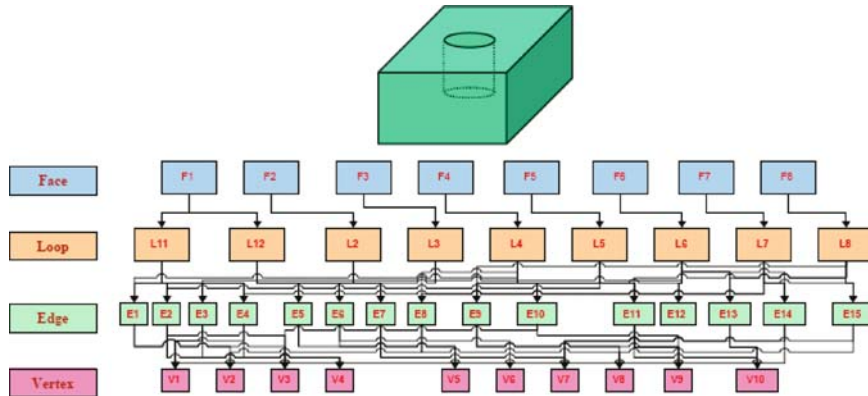


Fig. 6.2 A boundary representation (B-rep)

are broadly used to generate a solid model of a physical object [18]. The B-rep describes the geometry of an object in terms of its boundaries, which are the vertices, edges, and surfaces as shown in Fig. 6.2 [7]. In the next subsections, literature survey in the area of feature representations will be described.

### 6.3.1 Feature Representation by B-Rep

Tseng and Joshi [47] developed a method for feature recognition of mill-turned parts. B-rep was used to create rotational and prismatic components. This method was based on machining volume generation approach to recognize and classify features. The feature volumes were generated by sweeping boundary faces along a direction determined by the type of machining operations. In this approach, first, the part was segmented into several rotational machining zones. Next, the prismatic features were recognized on the bases of intermediate rotational shapes using volume decomposition and maximal volume sweeping and reconstruction. The classification of prismatic features was done by using the face adjacency relationships, while classification of rotational parts was performed by using profile edge patterns.

Aslan et al. [5] developed a feature extraction module for only rotational parts that are to be machined at turning centers. In this paper, the data interchange format (DXF) file was used to extract 2-D, which was represented by B-rep, for rotational parts. This extraction module was a part of an expert system called ASALUS. ASALUS was designed to manage the life cycle of rotational parts from design all the way to production by performing process planning using a generative approach and applying post-processing for two different computer numerical control (CNC) lathes. Prismatic features and the intersecting features were not involved in this module.

Nagaraj and Gurumoothy [34] described an algorithm to extract machinable volumes that need to be removed from a stock. This algorithm can handle both

prismatic and cylindrical components by using the B-rep model. The machinable volumes can be used to automate process planning and NC tool path generation. The algorithm identified the cavity volumes in the part with respect to the outermost faces in the part and filled them with the appropriate primitive volume to obtain the stock from which the part can be realized.

Kayacan and Celik [19] developed a feature recognition system for process planning for prismatic parts. This system was achieved with the B-rep modeling method to give vectorial direction knowledge and adjacent relationships of surface using the Standard for the Exchange of Product Model Data (STEP) standard interface program.

### ***6.3.2 Feature Representation by constructive solid geometry***

Requicha and Chan [37] developed a scheme for representing surface features in a solid modeler based on CSG and for associating tolerances and other attributes with such features. Their approach treats tolerances as attributes of object features which are part of the object's surface or topological boundary. They developed a graph structure called the variation graph to represent these features and attributes.

Shah and Roger [42] developed an integrated system for form features, precision features, and material features. The solid representation of the form features is stored as a feature producing volume (CSG tree) and Boolean operators. An object-oriented programming approach is used to represent the feature descriptions. The feature relationship graph is created at the top level of the model, where both the adjacency and the parent-child dependency of the form features are stored.

### ***6.3.3 Feature Recognition Techniques***

There are two approaches for building the CAD/CAM interface. They are design by features and feature recognition [14, 22]. Both approaches focus on the concept of "features." Design by features or the so-called feature-based design is a way of using design features to accomplish the construction of parts' CAD models. At first scene, this may seem to obviate the need for subsequent feature recognition. However, such features, being primarily design-oriented, have to be converted into manufacturing features to build the interface between feature-based design and CAM applications [41]. On the contrary, feature recognition is one of the major research issues in the area of automated CAD/CAM interface.

Various approaches and algorithms are proposed by many researchers. These are *syntactic pattern recognition* [45], *volume decomposition* [21], *expert system and logic* [33], the *CSG-based approach* [35], the *graph-based approach* [16], and the *neural-network-based approach* [11]. Most suggested methods for feature recognition used the internal representation of features that was created by the CAD system which had its own feature structure. On the contrary, there are few researches which

had used the standard format of product data. Some of them will be briefly discussed below.

Meeran and Pratt [30] used 2-D entities in a DXF file with no specific order and no relationship in their connectivity. The process of searching and sorting the entities is divided into three groups according to each of three orthographic views. The main objective of the approach was to recognize the machining features of prismatic parts that have planar or cylinder faces in terms of 2-D shapes which are located in the three orthographic views of the drawing. The first step of the approach is recognizing the isolated feature, then providing the library of patterns. The recognition is fundamentally based on the pattern matching approach. It cannot recognize prismatic parts and it has the drawback of using 2-D drawing.

Sheu [44] developed a CIM system for rotational parts. The parametric design and feature-based solid model were used to specify the manufacturing information required to the proposed system. In this system, the boundary of a solid model could be transferred directly into the line and arc profiles. The part model was created by using the cylinder, the cone, the convex arc, and the concave arc as primitives. This system had the ability to convert the wireframe part model into CSG representation by the feature recognition approach. Prismatic components and their interactions were not considered.

Ahmad and Haque [3] developed a feature recognition system for rotational components using DXF file. In this approach, the work geometric information of rotational parts is translated into manufacturing information through a DXF file. A feature recognition algorithm was used to recognize different features of the part from its DXF file, where geometric information of the part was stored after respective DXF codes. Finally, using the data extracted from DXF file, each feature of the part was recognized. The parts are symmetrical and were represented by two dimensions.

Mansour [28] developed simple software to link CAD packages and CNC machine tools. The main feature of this software was its ability to automatically generate part programs for machining sculptured surfaces. To achieve this objective, IGES files of simple or free-form surfaces were exploited. The data extracted from IGES files were used to graphically simulate the tool path due to the center and tip of a ball nose tool and a filleted cutter as well as the tip of a flat-end mill cutter. AUTOCAD and AUTOCAD development system (ADS) were used to convert trimmed surfaces to parametric spline surfaces. The package was not developed to handle other IGES entities which provide a wide range of free-form surfaces.

From the literature review, significant efforts have been focused on the development of fully automatic manufacturing feature recognition systems in the last two decades. No matter which approach is adopted, the geometric information always needs to be constructed early in order to advance the feature recognition. [Table 6.1](#) summarizes the literature review according to the feature recognition method, representation type, standard type, dimension type, and feature type. From the system implementation point of view, the best system should be independent of the format, that is, any kind of data format can be used for the input information and the internal geometric feature representation can be constructed on the basis of the input data.

**Table 6.1** Summary of literature review

Researcher	Feature recognition method	Representation type		Standard type			Dimension type			Feature type	
		CSG	B-rep	DXF	IGES	STEP	2-D	2.5-D	3-D	R	D
Liu <sup>25</sup>	Heuristic		*		*				*		*
FU <sup>12</sup>	Heuristic		*		*	*			*		*
Bhandarkar <sup>6</sup>	Heuristic		*			*			*		*
Roy <sup>40</sup>	AI		*						*	*	
Chang <sup>9</sup>	Syntactic		*				*			*	
Munns <sup>33</sup>	Logic	*							*		*
Liu <sup>24</sup>	Graph	*							*		*
Joshi <sup>16</sup>	Graph		*						*		*
Ciurana <sup>10</sup>	Vol. Dec.	*	*						*		*
Liu <sup>25</sup>	Heuristic				*				*		*
Madurai <sup>26</sup>	Expert		*							*	
Sharma <sup>43</sup>	AI					*			*		*
Linardkis <sup>23</sup>	Expert			*			*	*			*
Stalely <sup>46</sup>	Syntactic		*						*	*	
Vankataraman <sup>48</sup>	Graph		*						*		*
Zhao <sup>51</sup>	Heuristic								*		*
Chang <sup>8</sup>	AI								*		*
Kayacan <sup>19</sup>	Expert		*			*			*		*
Abouel Nasr <sup>2</sup>	Heuristic	*	*		*				*		*
Hwang <sup>15</sup>	Expert		*						*		*
Kao <sup>18</sup>	Graph		*						*		*
Zhang <sup>50]</sup>	Logic		*		*		*				*

R = rotational P = Prismatic vol.Dec = Volume decomposition AI = artificial Intelligent

B-rep is the most independent representation because other data formats have several limitations and disadvantages. Therefore, this chapter will adopt the solid modeling representation. On the contrary, the CSG technique will be used as a tool to construct the designed parts, so hybrid CSG/B-rep will be used in this research.

## 6.4 The Proposed Methodology

In this section, the part design is introduced through CAD software and is represented as a solid model by using the CSG technique as a design tool. The solid model of the part design consists of small and different solid primitives combined together to form the required part design. The CAD software generates and provides the geometrical information of the part design in the form of an ASCII file (IGES) [4] that is used as standard format which provides the proposed methodology the ability to communicate with the various CAD/CAM systems.

The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that is created specifically to extract the features from

the geometrical information on the basis of the geometric reasoning object-oriented approaches. The feature recognition program is able to recognize these features: slots (through, blind, and round corners), pockets (through, blind, and round corners), inclined surfaces, holes (blind and through), and steps (through, blind, and round corners). These features that are mapped to process planning as an application for CAM are called manufacturing information. Figure 6.3 shows the structure of the proposed methodology.

The proposed methodology presented in this chapter consists of three main phases: (1) a data file converter, (2) an object form feature classifier, and (3) a manufacturing features classifier as shown in Fig. 6.4. The first phase converts a CAD data in IGES/B-rep format into a proposed object-oriented data structure. The second phase classifies different part geometric features obtained from the data file converter into different feature groups. The third phase maps the extracted features to process planning point of view.

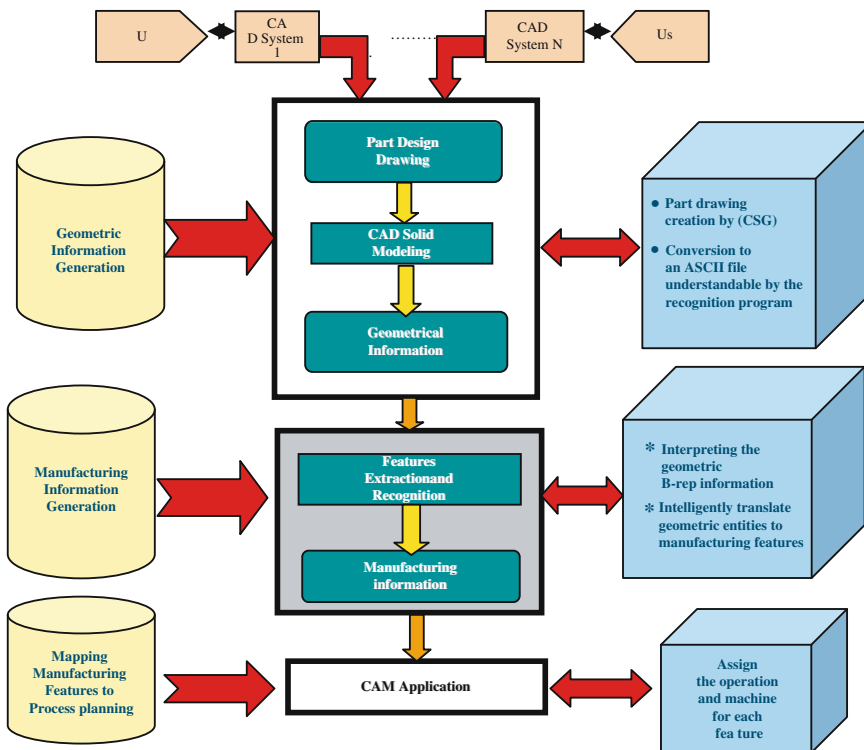


Fig. 6.3 Structure of the proposed methodology



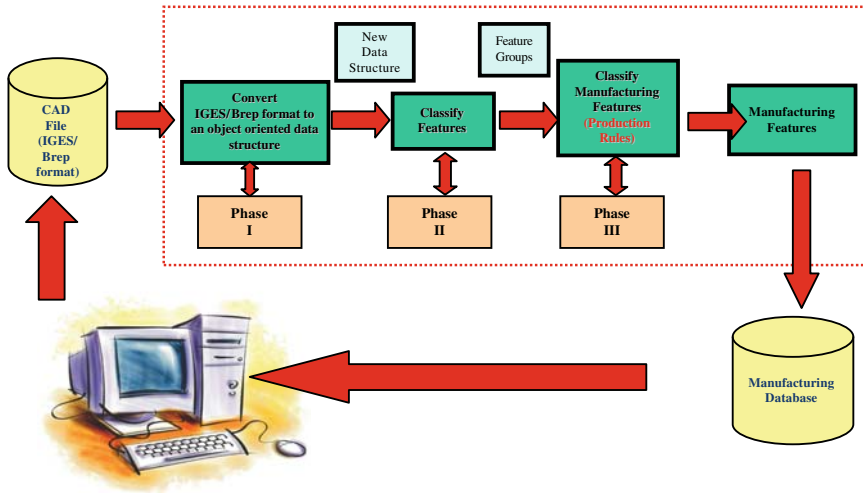


Fig. 6.4 Flowchart of extraction and classification of features

#### 6.4.1 Conversion of Computer-Aided Design Data Files to Object-Oriented Data Structure

IGES is a standard format that can be used to define the data of the object drawing in solid modeling CAD systems in B-rep structure (IGES 5.3, 1996). Object's geometric and topological information in IGES format can be represented by the entry fields that constitute the IGES file [17]. The geometric information are in low-level entities, such as lines, planes, circles, and other geometric entities for a given object, and the topological information that defines the relationships between the object's geometric parts are represented, for example, by the loops (external loop and internal loop). An external loop provides the location of main geometric profiles and an internal loop represents a protrusion or a depression (through/blind holes) on an external loop.

#### 6.4.2 The Overall Object-Oriented Data Structure of the Proposed Methodology

In order to have a good generic representation of the designed object for CAM applications, for example, process planning, the overall designed object description and its features need to be represented in a suitable structured database [25]. An object-oriented representation will be used in this chapter. The first step toward automatic feature extraction will be achieved by extracting the geometric and topological information from the (IGES/B-rep) CAD file and redefining it as a new object-oriented data structure as demonstrated in Fig. 6.5.

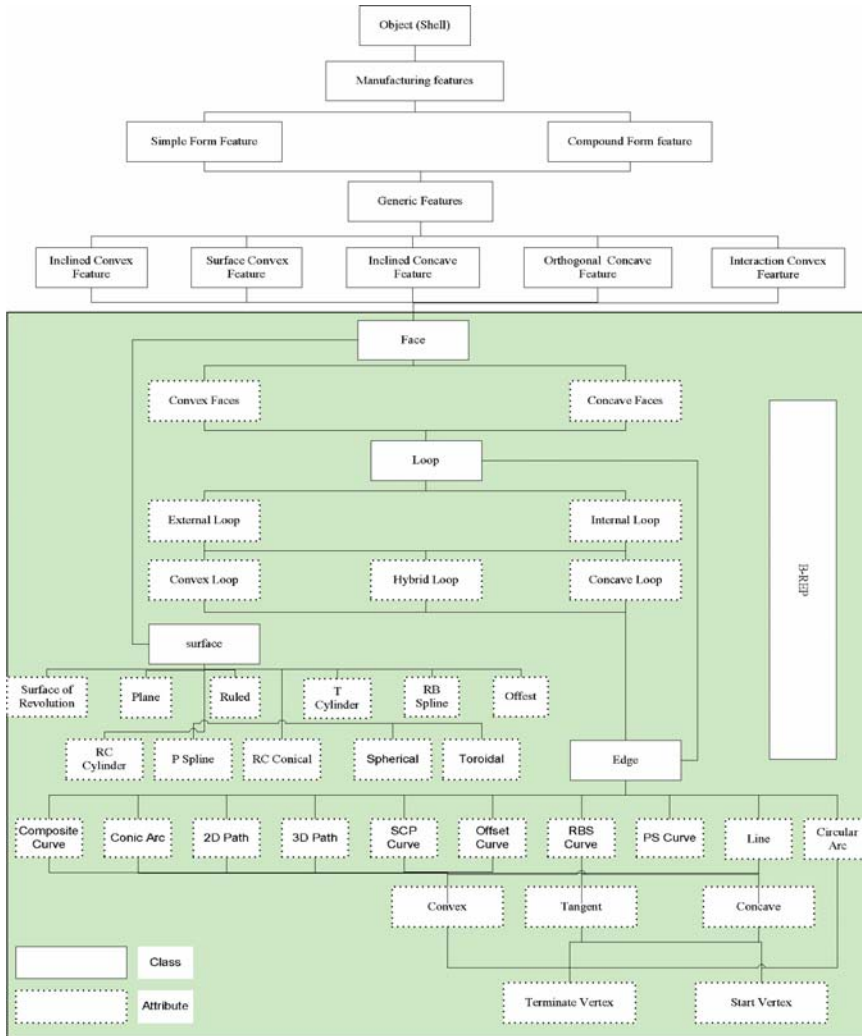


Fig. 6.5 Hierarchy of classes and attributes of the designed object

In this hierarchy, the highest level data class is the designed object (shell). An object consists of manufacturing features that can be classified into form features which decomposed of either simple or compound features. A simple feature is the result of two intersecting general geometric surfaces while compound feature is one that results from the interaction of two or more simple features (slot and pocket). Features are further classified into concave or convex as attributes in the generic feature class. Concave features consist of two or more concave faces, and convex features are decomposed of either one or more convex faces or the interaction between other features in the object. Moreover, faces of any designed part may be

any type of surfaces such as plane surface, ruled surface, and p-spline surface. Also, the edge of any designed part can be any type of edges such as line, circular arc, and conic arc.

Because of the attributes of the geometric and topological entities of form features (FF), form features can be classified into two categories [12], which are interior form feature ( $FF_{\text{interior}}$ ) and exterior form feature ( $FF_{\text{exterior}}$ ).  $FF_{\text{interior}}$  can be defined as the features which are located inside the basic face. On the contrary,  $FF_{\text{exterior}}$  can be defined as the features which are formed by the entire basic surface with its adjacent faces. The basic face can be defined as the face in which there are features located in that face. Moreover, the interior form features ( $FF_{\text{interior}}$ ) can be further classified into two low-level categories, convex interior form feature ( $FF_{\text{interior\_convex}}$ ) and concave interior form feature ( $FF_{\text{interior\_concave}}$ ).  $FF_{\text{interior\_convex}}$  is the convex section in a basic face, whereas  $FF_{\text{interior\_concave}}$  is the concave geometric section in the face.

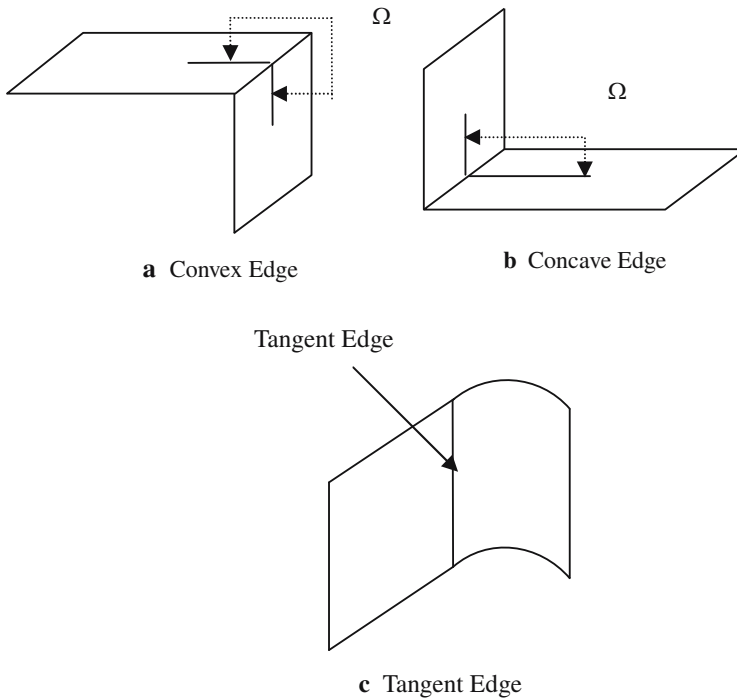
The basic idea to define concave features is to identify concave faces which are defined by a concave edge that connects two adjacent faces. A concave edge is determined by the concavity test that was adapted and modified from Liu et al. [25] and Hwang [15]. Basically, the edge is defined by two vertices (start vertex and terminate vertex) expressed in 3-D solid model in terms of coordinates ( $X, Y, Z$ ). On the contrary, convex features can be defined and classified as inclined, interaction, or surface. Inclined convex features are defined by a group of convex faces which are not parallel or perpendicular to the smallest surrounded envelope of the designed object. The other type of convex feature, interaction features, results from the interaction of two or more features. Surface convex features are features that locate on the exterior enclosing envelop.

#### 6.4.2.1 Classification of Edges

Edges forms the wireframe of any 3-D solid model and any two adjacent faces can be connected by one edge. Edges can be classified as concave, convex, or tangent edges, as shown in Fig. 6.6. This classification of edges will facilitate the implementation of the proposed methodology as proposed in Fig. 6.5. To represent edge types, the normal vectors of the two faces connected to that edge and the edge direction are determined. Then by applying the connectivity test, the edge classification can be achieved. Also, the angle between the two faces ( $\Omega$ ) is determined by the following equation:

$$\text{Angle}(\Omega) = \cos^{-1} \left( \frac{X_1 X_2 + Y_1 Y_2 + Z_1 Z_2}{\sqrt{X_1^2 + Y_1^2 + Z_1^2} \sqrt{X_2^2 + Y_2^2 + Z_2^2}} \right) \cdot \frac{\pi}{180} \quad (6.1)$$

where  $X_1, Y_1,$  and  $Z_1$  are the components of the normal vector  $N_1$  of the first face and  $X_2, Y_2,$  and  $Z_2$  are the components of the normal vector  $N_2$  of the second face.



**Fig. 6.6** Classifications of edges

#### 6.4.2.2 Classification of Loops

A loop can be defined as the boundary of a face. Moreover, it can be the intersection border of the face with its adjacent faces. In this research, a loop is used as a basic indication to recognize the interior and exterior form features. The loop can be classified as proposed in this research into two categories, external loops and internal loops [12, 49]. External Loop is the exterior border of a basic face of which the loop is examined, while internal loops are located within the basic face. By examining the basic face, an external loop can be identified by the maximum margin of the basic face and the internal loop can be recognized by the interior interface boundary of the basic face with its interior features.

#### 6.4.3 Definition of the Data Fields of the Proposed Data Structure

Generally, faces are the basic entities that constitute the features, which are further defined by edges that are represented in terms of vertices, which are defined in terms

of coordinates in CAD file. Therefore, the hierarchy of the designed object that was described in the previous section (Fig. 6.4) represents multilevel of different classes. All classes, except for the superclass representing an object as a whole, are objects of classes that are higher up in the data structure. For example, each edge object is represented in terms of vertex objects. Table 6.2 displays the data attributes required for each class in the object-oriented data structure that is defined before.

**Table 6.2** Definitions of classes and attributes

Class name	Attribute	Type
Point	X_ Coordinates	(Real)
	Y_ Coordinates	(Real)
	Z_ Coordinates	(Real)
Vertex	Inherits Point	Point
	Vertex_ID	(Integer)
Vertex_ List	Vertex_ Count	(Integer)
	Vertex_ List	(Vector of Vertex pointers)
Edge	Edge_ID	(Integer)
	Edge_Type	(Enumerated Constants)
	Start_Verx	(Vertex Pointer)
	Terminate_Verx	(Vertex Pointer)
	Concavity	(Enumerated Constants)
	Face_ Pointers [2]	(Array of Face Pointers)
	Loop_ Pointers [2]	(Array of Loop Pointers)
	Dimension	(real)
Edge_ List	Edge_ Count	(Integer)
	Edge_ List	(Vector of Edge Pointer)
Loop	Loop_ID	(Integer)
	Loop_ Concavity	(Enumerated Constants)
	Loop_ Type (External or Internal)	(Enumerated Constants)
	Edge_ List	(An edge list of loop edges)
	Face_ Pointers	(Pointer to face class)
Surface	Surface_Type	(Enumerated Constants)
Face	Face_ID	Number
	Surface_ Pointer	(Pointer to the Surface)
	External_ Loop	(Loop Pointer)
	Internal_ Loop_ Count	Number
	Internal_ Loop_ List	(Vector of Loop Pointers)
Shell	Vertex_ List	(Object of Vertex_ List class)
	Edge_ List	(Object of Edge_ List class)
	Loop_ List	(Vector of Loop Pointers)
	Surface_ List	(Vector of Surface Pointers)
	Face_ List	(Vector of Face Pointers)
	Name	(String)
	IGES_ File	(Object of IGES_ File Class)
Feature	Feature_ID	(Number)
	Feature_Type	(Enumerated constants)
	Feature_ Origin	(Vertex Pointer)
	Length	(Real)
	Width	(Real)

(continued)

**Table 6.2** (continued)

Class name	Attribute	Type
Compound Feature	Height	(Real)
	Radius	(Real)
	Face_List	(Vector of Face Pointers)
	Direction	(An object of Point Class)
	Feature_ID	(Number)
	Feature_Type	(Enumerated constants)
	Feature_List	(Vector of Feature Pointers)
	Merging_Feature	(Integer)

#### 6.4.4 Algorithms for Extracting Geometric Entities from CAD File

The IGES file is sequentially read (on a line basis) and parsed into appropriate entry classes known as **DEntry** and **PEntry** as the most important and useful sections of the IGES are the directory section and the parameter section. **DEntry** represents an entry in the directory section, while **PEntry** represents an entry in the parameter section. The collection of directory entry classes is contained in a container class called **DSection**.

Similarly, the parameter entry classes are contained in the **PSection** class. A **Parser class** object is created using these classes to parse the information present in the entries and classify the information into different classes that are used to represent different entities of the diagram described by the IGES file. Two algorithms for extraction of data from the IGES file into a proper set of data structures were developed. The first algorithm is developed for extracting entries from directory and parameter sections of the IGES file and it will be addressed later in this chapter. On the contrary, the second algorithm is developed for extracting the basic entities of the designed part [1].

##### 6.4.4.1 Algorithm for Extracting Entries from Directory and Parameter Sections

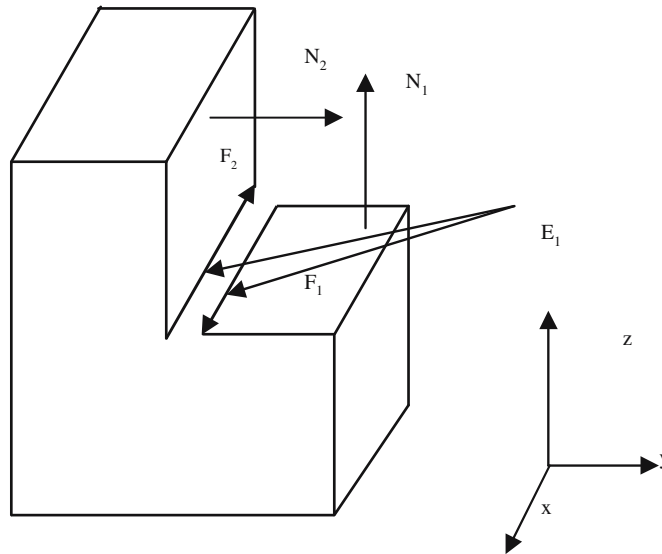
- ```
// Algorithm to extract the directory entries
// and the parameter section entries from the iges file.
// This process takes place during the construction of an object of IGESFile class.
// Each such object represents one IGES file.
```
1. Create a file descriptor IgesFile.
  2. Create an empty dSection1 class (container to store dEntry objects).
  3. Create an empty pSection1 class (container to store pEntry objects).
  4. Open the Iges file for reading using IgesFile file descriptor // Read the file to scan and extract the directory and parameter sections.
  5. While ReadLine line1 from the Igesfile

- 5.1 If line1 belongs to Directory section
  - 5.1.1 If line1 is the first line of Dsection
    - 5.1.1.1 Set dIndex to 1
  - 5.1.2 ReadLine line2 from the Igesfile
  - 5.1.3 Create an object dEntry1 of class DEntry
  - 5.1.4 Set dEntry1 index using dIndex
  - 5.1.5 Initialize dEntry1 using string Line1 + Line2
  - 5.1.6 Add dEntry1 to dSection1 class
  - 5.1.7 Set dIndex = dIndex + 1
- 5.2 If line1 belongs to Parameter Section
  - 5.2.1 If line1 is the first line of PSection
    - 5.2.1.1 Set pIndex to 1
  - 5.2.2 Create an empty string Line2
  - 5.2.3 while pEntry data incomplete
    - 5.2.3.1 ReadLine Line3 from the Igesfile
    - 5.2.3.2 Append Line3 to Line2
  - 5.2.4 Create an object pEntry1 of class PEntry
  - 5.2.5 Set pEntry1 index equal to pIndex
  - 5.2.6 Initialize pEntry1 using string Line1 + Line2
  - 5.2.7 Add pEntry1 to pSection1 class
  - 5.2.8 Set pIndex = pIndex + 1
- 5.3 If line1 belongs to Terminate Section
- 5.4 exit while loop
- 6. End of while loop

#### **6.4.5 *Extracting Form Features from Computer-Aided Design Files***

The edge direction and the face direction are the basic entities information that is used to extract both simple and compound form features from the object data structure. The edge directions in object models can be defined such that when one walks along an edge, its face is all the time on the left-hand side. When an edge is located in the external loop of a face, its direction will be in anticlockwise direction relative to the surrounding face. On the contrary, when an edge is located in the internal loop of a face, its direction will be clockwise [15].

The concave edge test used in research is based on the cross product of the normal vectors of the two faces joined by a given edge. This is done by applying vector geometry to the face and edge direction vectors. [Figure 6.7](#) shows the symbols used in this test where the  $i$ th face is designated as  $F_i$ , its corresponding normal direction vector is defined as  $N_i$  in the upward direction with respect to the given face, and the  $k$ th edge is designated as  $E_k$ . For each edge of the designed part, the edge ( $E_k$ ) is shared by two faces ( $F_i$  and  $F_j$ ) where the order is right to left from the left side



**Fig. 6.7** A concave edge example

of the edge view perspective. The direction vectors of the faces are as described above ( $N_i$  and  $N_j$ ). Finally, the edge's directional vector is given with respect to the face  $F_i$  using the loop  $L_i$  that contains the edge ( $E_k$ ). The following is the methodology for the concavity test:

1. The cross (vector) product ( $V$ ) of the directional vectors of the faces is determined as follows:

$$V = N_i \times N_j$$

2. The direction of the edge  $E_k$  with respect to the face  $F_i$  is determined. The normal vector  $N_i$  of face  $F_i$  must be the first component in the cross product of step 1.
3. If the direction vector of the edge  $E_k$  from step 2 is in the same direction of cross product  $V$ , then the edge  $E_k$  is a convex edge that concludes  $F_i$  and  $F_j$  are convex faces, otherwise, it will be a concave edge and  $F_i$  and  $F_j$  are concave faces. Also, if the cross product vector  $V$  is a zero vector that means that the edge is of tangent category.

This procedure will be done on all the edges of the object to define the concave, tangent, or convex faces. Moreover, concave features will be identified by the premise that concave faces include at least one concave edge with adjacent concave faces forming a concave face set. Each concave face set defines a concave feature. Similarly, adjacent convex faces form a convex face set. Two algorithms are developed to determine the concavity of both edges and loops. The concavity edge algorithm will be discussed later in this chapter while the concavity loop algorithm can be found in Abouel Nasr and Kamrani [2].



#### 6.4.5.1 Algorithm for Determination the Concavity of the Edge

The following algorithm is used to find the concavity of a line entity. This algorithm is used by the Line class to find the entities.

1. Length of Line =  $\sqrt{[(\text{startX-termX})^2 + (\text{startY-termY})^2 + (\text{startZ-termZ})^2]}$
2. concavity = UNKNOWN
3. If face1 surface type == PLANE and face2 surface type == PLANE
  - 3.1. Assign crossDir = cross product of face1 normal vector and face2 normal vector
  - 3.2. If crossDir == 0
    - 3.2.1 concavity = TANGENT
  - 3.3 Else
    - 3.3.1 Calculate the direction vector edgeDir for the line with respect to the loop
      - 3.3.1.1 If crossDir is in the same direction as edgeDir
        - 3.3.1.1.1 concavity = CONVEX
        - 3.3.1.1.2 Else
          - 3.3.1.1.2.1 concavity = CONCAVE
4. If face1 surface type == RCCSURFACE and face2 surface type == PLANE
  - 4.1 Find dir1 (direction) that is orthogonal to the plane containing the edge and the axis of face1
  - 4.2 If dir1 and normal of face2 are orthogonal to each other
    - 4.2.1 concavity = TANGENT

#### 6.4.5.2 Algorithms for Feature Extraction (Production Rules)

The proposed methodology is able to extract many manufacturing features. Each feature has its own algorithm (production rule). The following are the two algorithms used for extraction of both step through and step through with round corners:

Feature: **STEP THROUGH** (Fig. 6.8)

1. For every concave edge of type Line in the edge list.
2. If the two common faces (face1 and face2) of the edge (e1) are **plane** and **orthogonal** to each other
  - 2.1. if outer loop concave edge count equals 1 in both the faces
    - 2.1.1. STEP THROUGH found
    - 2.1.2. Create a new StepT object and add to feature list
3. End For

Feature **STEP THROUGH ROUND CORNER** (Fig. 6.9)

1. For every 3 tangent edges of type Line in the edge list ( $e_1, e_2, e_3$ )
2. If the common face of the 2 edges is quarter cylindrical surface ( $F_2, F_3$ ).
  - 2.1 The other two faces ( $F_1, F_4$ ) connected to edges are perpendicular to each other.
    - 2.1.1. If concave edge count of the outer loops of the four faces equals 0 each.

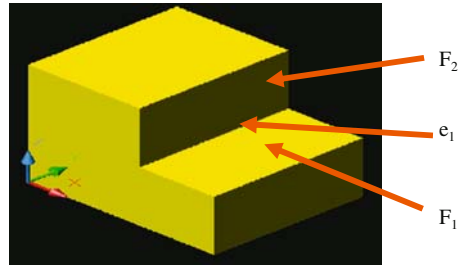


Fig. 6.8 Step through

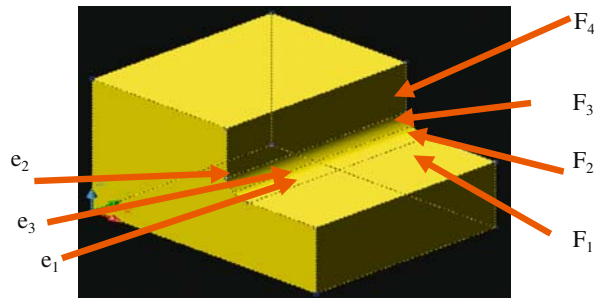


Fig. 6.9 Step through round corner

2.1.1.1. STEP THROUGH ROUND CORNER found

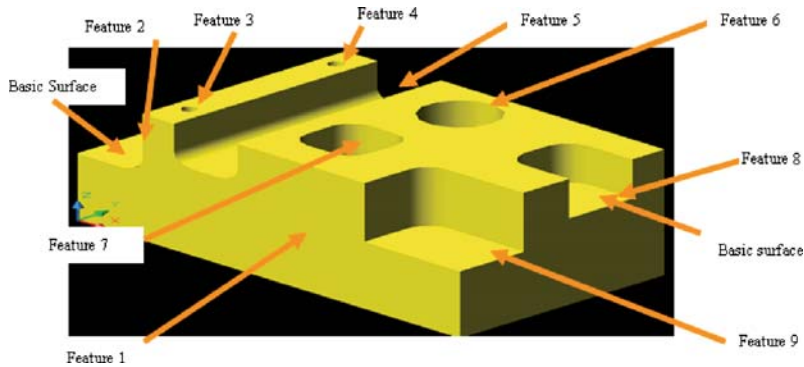
2.1.1.2. Create a new StepT\_RC object and add to feature list.

3. End For

## 6.5 Illustrative Example

The proposed methodology is used for the component illustrated in Fig. 6.10. Mechanical Desktop 6 Power Pack is the CAD system used that supports B-rep and IGES translator. However, other similar CAD systems that support IGES translator can be used. The proposed methodology is developed by using Microsoft Visual C++ 6 windows based on PC environment. The designed object consists of eight different features and prismatic raw material.

Mechanical Desktop CAD system is one of the recent CAD softwares that can be used for design applications. This CAD system supports both B-rep and IGES translator (version 5.3). In Mechanical Desktop, the designed part can be represented by 2-D or 3-D drawings. In this chapter, the part designs are represented by 3-D solid models using the CSG technique as a design tool. After creating the 3-D solid model of the designed part, the CAD user has to export the 3-D solid model



**Fig. 6.10** An illustrative example

file into the IGES format provided that B-rep option version 5.3 should be highlighted. Then the user has to save the IGES file in order to be used as an input for the developed program. This file is designated as geometrical attributes file. By applying the proposed methodology, the final results are shown in [Table 6.3](#).

## 6.6 Conclusion

In this chapter, a new methodology for extraction manufacturing features from IGES file format was introduced. The proposed methodology was developed for 3-D prismatic parts that were created by using solid modeling package by using the CSG technique as a drawing tool. The system takes a neutral file in IGES format as input and translates the information in the file to manufacturing information. The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that was created specifically to extract the features from the geometrical information on the basis of the geometric reasoning and object-oriented structure approaches.

The methodology discussed has several advantages over other methods suggested in the literature. First, by using the object-oriented approach, the proposed methodology has the ability to provide a good and generic representation of the simple and compound product data in which the feature, geometry, topology, and manufacturing data are associated. Second, the proposed methodology is flexible to the variations of the IGES file format from different vendors that offer different CAD systems. Third, the proposed methodology separated the extraction module of the IGES entities of the designed part from the subsequent modules of the program. This makes the proposed methodology easily adaptable to any other standard format such as STEP or DXF. Fourth, the proposed methodology is using the IGES format as a standard input. Most feature recognition approaches found in the

**Table 6.3** Manufacturing features

| Feature ID | Feature type           | Faces ID                                      | Edges ID                                             | Location               | Feature name                  | Dimension |    |       |
|------------|------------------------|-----------------------------------------------|------------------------------------------------------|------------------------|-------------------------------|-----------|----|-------|
|            |                        |                                               |                                                      |                        |                               | L         | W  | H     |
| [1]        | Prismatic              | [38][50][48][39][42][49]                      | [120][125][137]                                      | [84] = (0,0,0)         | Raw material                  | 60        | 32 | 32    |
| [2]        | FF <sub>exterior</sub> | [41][2][40][42]                               | [7][5][126]                                          | [6] = (9,12,0,17.88)   | Step_Through (Round Corner)   | 32        | 12 | 7     |
| [3]        | FF <sub>interior</sub> | [39][48]                                      | [42][53]                                             | [26] = (12,5,28,0)     | Hole_Through                  |           |    | 32    |
| [4]        | FF <sub>interior</sub> | [39][48]                                      | [43][54]                                             | [26] = (12,5,5,0)      | Hole_Through                  |           |    | 32    |
| [5]        | FF <sub>exterior</sub> | [36][33][4][34][3][35][37]                    | [103][13][15][11][9][108]                            | [13] = (15,88,0,22.88) | Slot_Through (Round Corner)   | 32        | 7  | 4     |
| [6]        | FF <sub>interior</sub> | [32][17]                                      | [33][17]                                             | [26] = (35,25,17)      | Hole_Blind                    |           |    | 32    |
| [7]        | FF <sub>interior</sub> | [19][10][18][21][9][20][23][8][22][25][7][24] | [63][39][37][60][35][33][66][31][29][71][27][25]     | [57] = (28,8,0)        | Pocket_Through (Round Corner) | 32        | 4  | 4     |
| [8]        | FF <sub>interior</sub> | [30][29][28][5][27][26][6][31]                | [85][17][19][80][21][23][87][84][20][82][79][24][90] | [61] = (50,20,22)      | Slot_Blind (Round Corner)     | 7         | 4  | 16.49 |
| [9]        | FF <sub>interior</sub> | [44][43][1][46][45]                           | [132][1][3][135][131]                                | [2] = (45,88,9,12,17)  | Step_Blind (Round Corner)     | 15        | 12 | 7     |

literature have been developed by the internal data structure of the CAD system. Therefore, these approaches are specific domain for these CAD systems. The proposed methodology can also extract simple curved features like round corners. Most researchers who extracted curved features tried to approximate the curvature of the surfaces which result in inaccurate representations or extraction of the manufacturing features. Finally, the proposed methodology operates in 3-D solid modeling environment which gives it a powerful ability to be used by the current manufacturing technology.

## References

1. Abouel Nasr, E., & Kamrani, A., A Feature Interaction Approach for CAM Applications, 37th International Conference on Computers and Industrial Engineering, Alexandria, Egypt (2007).
2. Abouel Nasr, E., & Kamrani, A., A New Methodology for Extracting Manufacturing Features from CAD System, International Journal of Computers and Industrial Engineering, 15(1), 389–415 (2006).
3. Ahmad, N., & Haque, A., Manufacturing Feature Recognition of Parts Using DXF Files, Fourth International Conference on Mechanical Engineering, Dhaka, Bangladesh (1), 111–115 (2001).
4. ANS US PRO/IPO-100, Initial Graphics Exchange Specifications: IGES 5.3 (1996).
5. Aslan, E., Seker, U., & Alpdemir, N., Data Extraction from CAD Model for Rotational Parts to be Machined at Turning Centers, Turkish Journal of Engineering and Environmental Science, 23(5), 339–347 (1999).
6. Bhandarkar, M.P., Downie, B., Hardwick, M., & Nagi, R., Migrating from IGES to STEP: One to One Translation of IGES Drawing to STEP Drafting Data, Computers in Industry, 41(3), 261–277 (2000).
7. Chang, T.C., Expert Process Planning for Manufacturing, Addison-Wesley Publishing Company, Reading, MA (1990).
8. Chang, H.C., Lu, W.F., & Liu, F.X., Machining Process Planning of Prismatic Parts Using Case-Based Reasoning and Past Process Knowledge, Applied Artificial Intelligence, 16(4), 303–331 (2002).
9. Chang, P., & Chang, C., An Integrated Artificial Intelligent Computer Aided Process Planning System, International Journal of Computer Integrated Manufacturing, 13(6), 483–497 (2000).
10. Ciurana, J., Romeu, M.L.G., & Castro, R., Optimizing Process Planning using Groups of Precedence Between Operations Based on Machined Volume, Engineering Computations, 20(1/2), 67–81 (2003).
11. Devireddy, C.R., & Ghosh, K., Feature-Based Modeling and Neural Networks-Based CAPP for Integrated Manufacturing, International Journal of Computer Integrated Manufacturing, 12(1), 61–74 (1999).
12. Fu, M.W., Lu, W.F., Ong, S.K., Lee, I.B.H., & Nee, A.Y.C., An Approach to Identify Design and Manufacturing Features from a Data Exchanged Part Model, Computer Aided Design, 35(11), 979–993 (2003).
13. Groover, M.P., Automation, Production Systems, and Computer-Integrated Manufacturing, Prentice-Hall, Inc., New Jersey (2001).
14. Harun, W.A., & Case, K., Feature-Based Representation for Manufacturing Planning, International Journal of Production Research, 38(17), 4285–4300 (2000).
15. Hwang, J., Rule-Based Feature Recognition: Concepts, Primitives and implementation. MS Thesis, Arizona State University (1988).

16. Joshi, S., & Chang, T.C., Graph-Based Heuristics for Recognition of Machined Features from 3D Solid Model, *Computer Aided Design*, 20(2), 58–66 (1988).
17. Kahrs, M., The Heart of IGES, *Software-Practice and Experience*, 25(8), 935–946 (1995).
18. Kao, C.-Y., Kumara, S.R.T., & Kasturi, R., Extraction of 3D Object Features from CAD Boundary Representation using Super Relation Graph Method, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(12), 1228–1233 (1995).
19. Kayacan, M.C., & Celik, S.A., Process Planning System for Prismatic Parts, *Integrated Manufacturing Systems*, 14(2), 75–86 (2003).
20. Lee, K., *Principles of CAD/Cam/CAE Systems*, Addison-Wesley Publishing Co., Inc., Boston, MA (1999).
21. Lin, A.C., & Lin, S.-Y., Volume Decomposition Approach to Process Planning for Prismatic Parts with Depression and Protrusion Design Features, *International Journal of Computer Integrated Manufacturing*, 11(6), 548–563 (1998).
22. Lin, A.C., Lin, S.-Y., & Cheng, S.-B., Extraction of Manufacturing Features from a Feature-Based Design Model, *International Journal of Production Research*, 35(12), 3249–3288 (1997).
23. Linardakis, S., & Mileham, A.R., Manufacturing Feature Identification for Prismatic Components from CAD DXF Files, *Advances in Manufacturing Technology VII, Proceedings of 9th National Conference on Manufacturing Research*, 37–41 (1993).
24. Liu, C.-H., Perng, D.-B., & Chen, Z., Automatic Form Feature Recognition and 3D Part Recognition from 2D CAD Data, *Computer and Industrial Engineering*, 26(4), 689–707 (1994).
25. Liu, S., Gonzalez, M., & Chen, J., Development of an Automatic Part Feature Extraction and Classification System Taking CAD Data as Input, *Computers in Industry*, 29(3), 137–150 (1996).
26. Madurai, S.S., & Lin, L., Rule-Based Automatic Part Feature Extraction and Recognition from CAD Data, *Computers and Industrial Engineering*, 22(1), 49–62 (1992).
27. Ma, X., Zhang, G., Liu, S., & Wang, X., Measuring Information Integration Model for CAD/CMM, *Chinese Journal of Mechanical Engineering*, 16(1), 59–61 (2003).
28. Mansour, S., Automatic Generation of Part Programs for Milling Sculptured Surfaces, *Journal of Materials Processing Technology*, 127(1), 31–39 (2002).
29. Marri, A., & Kobu, B., Implementation of Computer-Integrated Manufacturing in Small and Medium Enterprises, *Industrial and Commercial Training*, 35(4), 151–157 (2003).
30. Meeran, S., & Pratt, M.J., Automatic Feature Recognition from 2D Drawings, *Computer Aided Design*, 25(1), 7–17 (1993).
31. Meeran, S., Taib, J.M., & Afzal, M.T., 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), 465–495 (2003).
32. Miao, H.K., Sridharan, N., & Shah, J., CAD/CAM Integration Using Machining Features, *International Journal of Computer Integrated Manufacturing*, 15(4), 296–318 (2002).
33. Munns, A., Li, Y., & Wang, X.C., A Rule-Based Feature Extraction from CSG Representations and an Application in Construction, *Proceedings of SPIE—The International Society of Optical Engineering*, 2620(1), 269–276 (1995).
34. Nagaraj, H.S., & Gurumoorthy, B., Machinable Volume Extraction for Automatic Process Planning, *IIE Transactions*, 34(4), 393–410 (2002).
35. Natekar, D., Zhang, X., & Subbarayan, G., Constructive Solid Analysis: A Hierarchical, Geometry-Based Meshless Analysis Procedure for Integrated Design and Analysis, *Computer Aided Design*, 36(5), 473–486 (2004).
36. Reiter, W.F., Collaborative Engineering in the Digital Enterprise, *International Journal of Computer Integrated Manufacturing*, 16(7–8), 586–589 (2003).
37. Requicha, A.A.G., & Chan, S.C., Representation of Geometric Features, Tolerances, and Attributes in Solid Modelers Based on Constructive Geometry, *IEEE Journal of Robotics and Automation*, RA-2(3), 156–166 (1986).
38. Rouibah, K., & Casekey, K.R., Change Management in Concurrent Engineering from a Parameter Perspective, *Computers in Industry*, 50(1), 15–34 (2003).

39. Roucoules, L., Salomons, O., & Paris, H., Process Planning as an Integration of Knowledge in the Detailed Design Phase, *International Journal of Computer Integrated Manufacturing*, 16(1), 25–37 (2003).
40. Roy, U., & Liu, C.R., Feature-Based Representational Scheme of a Solid Modeler for Providing Dimension and Tolerancing Information, *Robotics & Computer-Integrated Manufacturing*, 4(3/4), 335–345 (1998).
41. Rozenfeld, H., & Kerry, H.T., Automated Process Planning for Parametric Parts, *International Journal of Production Research*, 37(17), 3981–3993 (1999).
42. Shah, J.J. & Roger, M.T., Expert Form Feature Modeling Shell, *Computer Aided Design*, 20(9), 515–524 (1988).
43. Sharma, R., & Gao, J.X., Implementation of STEP Application Protocol 224 in an Automated Manufacturing Planning System, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 216(1), 1277–1289 (2002).
44. Sheu, J.J., A Computer Integrated Manufacturing System for Rotational Parts, *International Journal of Computer Integrated Manufacturing*, 11(6), 538–547 (1998).
45. Sreevalsan, P.C., & Shah, J.J., Unification of Form Feature Definition Methods, *Proceedings of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design*, 83–106 (1992).
46. Staley, S.M., Henderson, M.R., & Anderson, D.C., Using Syntactic Pattern Recognition to Extract Feature Information from a Solid Geometric Database, *Computers in Mechanical Engineering*, 2(2), 61–66 (1983).
47. Tseng, Y.J., & Joshi, S.B., Recognizing of Interacting Rotational and Prismatic Machining Features from 3D Mill-Turn Parts, *International Journal of Production Research*, 36(11), 3147–3165 (1998).
48. Venkataraman, S., Sohoni, M., & Kulkarni, V., A Graph-Based Framework for Feature Recognition, Appearing in *ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, 194–205 (2001).
49. Zeid, I., *Mastering CAD/CAM*. McGraw-Hill Higher Education (2004).
50. Zhang, G.X., Liu, S.G., Ma, X.H., & Wang, Y.Q., Toward the Intelligent CMM, *CIRP Annuals, Manufacturing Technology*, 51(1), 437–442 (2002).
51. Zhao, Z., Ghosh, S.K., & Link, D., Recognition of Machined Surfaces for Manufacturing Based on Wireframe Models, *Journal of Materials Processing Technology*, 24(1), 137–145 (1990).