Jurnal Teknologi, bil. 26, Jun 1997 hlm. 65–74

BOUNDARY REPRESENTATION-BASED FEATURE RECOGNITION

NAPSIAH BT ISMAIL & NOOH BIN ABU BAKAR
Business and Advanced Technology Centre (BATC)
University Technology Malaysia
Kuala Lumpur, Malaysia

Abstract. This paper introduces an ongoing research which is aimed at the development of an intelligent form feature extraction system from Computer Aided Design (CAD) database, a high level data structure form useful for Computer Aided Manufacturing (CAM) such as Automated Process Planning System (APPS). Part description in CAD models is the form of basic geometry and topology that is unsuitable for direct application in APPS. Furthermore, CAD software does not incorporate sufficient manufacturing specific data to be used in APPS. Therefore, feature recognition systems will provide the capabilities for bridging the gap between the CAD database and the CAM database. A solid boundary representation (B-rep) model of the part is used to describe the part. This paper concentrates on the recognition of machinable features of either depression or protrusion types to be used in Automated Process Planning System. Logical procedures were developed to recognise these features which consists of both simple and intersecting features.

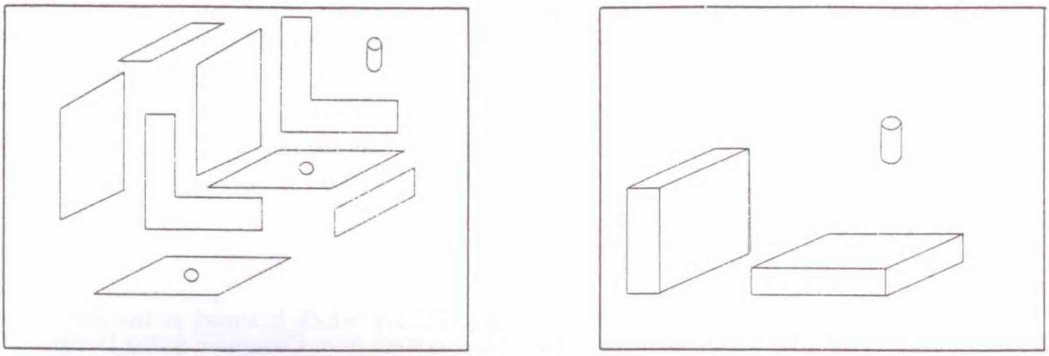
1 INTRODUCTION

One of the major challenges of Computer Integrated Manufacturing (CIM) is to integrate Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) databases. CAD representations of a model stored in computer memory is low level geometric entities such as faces, lines and points. However, in CAM, decisions are mainly made based on physical attributes such as holes, grooves and pockets which are commonly called features. Therefore, an automatic and intelligent recognition procedures are required to transform the low level data into higher level data (i.e machinable features) useful for manufacturing activities. Once a feature is recognised, the machining operations and other machining data associated with the identified feature could be extracted and passed to another processor so as to generate a process plan. Solid models (i.e. Boundary representation (B-rep) and Constructive Solid Geometry (CSG)) have been gaining wider acceptance in modelling the part to be used for automated feature recognition and extraction systems.

Solid models contain both the geometric data and topology information of the corresponding objects. Geometry is concerned with the equations of surfaces and position of the vertices while topology is the connectivity and associativity of the object (i.e. relationship between the faces, edges and vertices). B-rep is a method of representing a solid object using its bounding surfaces whilst CSG is based on a collection of primitive solids (i.e. cubes, cylinders, etc) (see Figure 1).

The data structure of CSG is a binary tree called the CSG tree. The CSG tree stores a model in an implicit form; the edges and vertices of the final part must be calculated from the set operations on the primitives. Features, however, may be represented explicitly as in

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX



a) Boundary representation (B-rep) b) SCG representation

Fig. 1 Types of Solid Modelling Representations

subtracting a cylinder to create a hole. The advantage of using B-rep stems from the fact that the model explicitly represents the final part. The faces, edges, vertices are represented exactly as in the final part. B-rep seems to be directly usable for CAM since manufacturing process deals with surfaces, e.g. machining operations generate surfaces. The information required for manufacturing can be extracted easily from an explicit surface definition [8].

2 EXTRACTION OF FORM FEATURES

Various approaches have been employed to automatically recognise and extract features both in B-rep and CSG representation such as pattern matching, rule-based, graph-based theory, volume decomposition from CAD systems but success has been limited to simple parts geometry. In B-rep, algorithms were based on direct matching of topologies/geometry templates [1,7], syntactic pattern recognition [10,14], graph-based approaches [2,6,8,11], expert system and logic approach [16] and neural network [12]. Choi [1] used pattern matching techniques to recognise polyhedral and non-polyhedral features. This method is very dedicated and recognition algorithm is highly feature dependent. As an example, a cylindrical protrusion may be recognised as a hole. Pattern matching is complicated especially for 3D complex shapes (i.e. intersecting features).

In syntactic pattern recognition, a picture is represented by some semantic primitives written in a picture language. A set of grammars consisting of some re-write rules define a particular pattern. A parser is then used to apply the grammar to the picture. If the syntax of the picture matches the grammar, then the picture can be classified as belonging to the particular pattern class.

A graph-based approach requires matching the feature graph to the appropriate subgraph in a solid modeller database. Joshi and Chang [8] represent parts by transforming B-rep into Attributed Adjacency Graph (AAG) with faces as nodes, and edges as arcs with assigned attributed values. Only polyhedral features with planar surfaces have been discussed and features like holes, bosses (non-planar surfaces) are not mentioned. The graph-based approaches have the advantage of being more easily formalised and thus analysed. The disadvantages of graph-based techniques are that they have high computational complexity and restricted to predefined feature pattern graph and cannot recognise general features of varying topology, for example n-sided pocket. Furthermore, different features

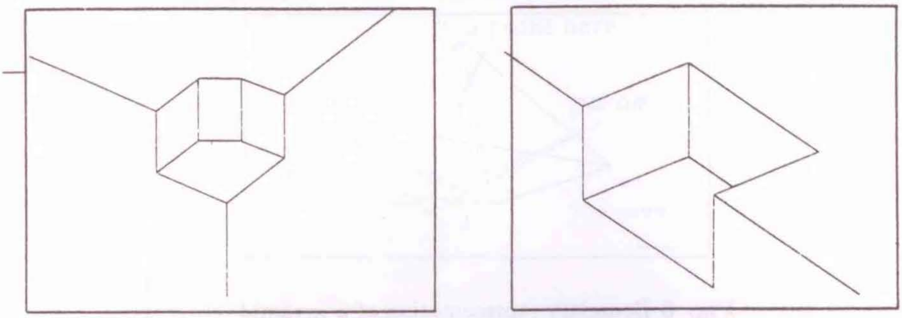


Fig. 2 Features that have the same graph

might have the same graph (same topological relationship - see Figure 2) that need extra computation to differentiate between them.

Prabhakar [12] employed neural net technique and only limited to simple features. No interaction features can be recognised. Tseng [15] used B-rep part presentation for multiple interpretation of features so that a component can be machined in multiple ways. Only pockets features type were considered.

The extraction in a CSG is much more complicated compared to B-rep because of the non-uniqueness of the representation of the CSG tree [4,9] such as that the same part may be created with different sets of primitive volumes. Kim [9] used Alternating Sum of Volume (ASM). The features considered are depression features and protrusion features.

B-rep based feature recognition methods have suffered from a lack of robust algorithms, particularly when feature interactions are present. This paper describes a research in progress on feature recognition procedures to extract form features of a part from B-rep created by AutoCAD and attempts to overcome some of the limitation mentioned above. Any other solid modeller capable of providing a B-rep can be used. AutoCAD was chosen as the CAD modeller because it is widely used in industry especially in small and medium scale industries. To develop the algorithm for feature recognition, the logic approach was implemented using a user interface of AutoCAD, namely AutoLISP language. The recognition system has been tested for holes and bosses features. Intersecting feature can also be recognised.

3 PART REPRESENTATION SCHEME

B-rep models contain sufficient information about faces, edges and vertices, which is defined as boundary of solid, $BS = (F, E, V)$ where $F = \{\text{set of faces}\}$, $E = \{\text{set of edges}\}$ and $V = \{\text{set of vertex}\}$. The model obeys Euler's rule. For polyhedral without holes, the number of edges must always be less than the sum of the numbers of faces and vertices where $F + V = E + 2$ (e.g. a pyramid, for instance, has 5 faces, 5 vertices and 8 edges: $5 + 5 = 8 + 2$).

B-reps are based on the observation that solids can be considered as being bounded by a number of faces, which are bounded by a number of edges, which in turn are bounded by two vertices such as in Figure 3. A face, which may be planar or curved, can be represented by the equation of the surface. An edge can be the intersection of two faces. An edge can be a line segment, arc or circle. Each vertex is defined by (x, y, z) coordinates.

The features considered in this research are classified as protrusion and depressions type of 2.5D features because they can be defined by sweeping a 2D cross section along a trajec-

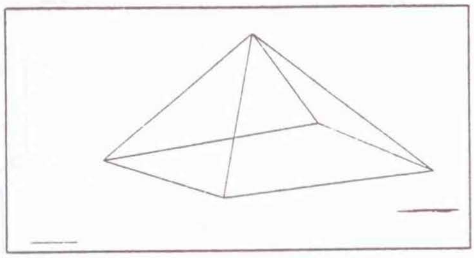


Fig. 3 Boundary representation of a pyramid

tory. The features are formed by a set of faces grouped together to form functional purposes (e.g. machinable features). The faces type are planar and curvilinear.

4 FEATURE RECOGNITION SYSTEM

In order to retrieve the relevant entities defining the geometric parts only, a program was developed using user interface language of AutoCAD, namely AutoLISP language. These data are then used as an input for the feature recognition system. Recognition process starts with topological checking (the number of faces and its associated edges) from the B-rep. A new approach using boundary point classification is tested to determine its suitability for feature recognition algorithm.

4.1 Feature recognition procedure

In this section, logic procedures method for feature recognition is described. The implementation of the system can be divided into seven steps:

(1) Processing the geometric data (entities in CAD model geometry).
(2) Identifying all faces together with their face-information.
(3) Determining the edges bounding a face and their *face-loop* from edge information list.
(4) Identifying the relationship between faces. This is important in process planning since it affects the tool access directions, operation sequencing and set-up strategy.
(5) Determining points and their classification on the *face-loop*.
(6) Extracting recognised features from the database. This function specifies the types of features found, e.g. holes, bosses.
(7) Determining the features parameters, e.g. diameters (radii), length, height.

Each feature can be defined by a specific combination of faces, edges and vertices. As an example, a cylindrical hole or a boss can be recognised by the existence of cylindrical face and its associated circular edges. A hole can be differentiated from a boss by determining whether a point inside the boundary lies inside the solid, outside the solid or on the solid. The point must be taken near the boundary of the circular edges to avoid misinterpretation if a hole is inside the boss (see Figure 4). Since holes or bosses have two circular edges, the point near the boundary of circular edge, at *edge1* and *edge2* are defined as *class1* and *class2* respectively. Further classification is needed to determine either a point which lies inside the solid, on the solid and outside the solid. If the point lies in the solid, on the solid or outside the solid, the classification are defined as "*inobject*", "*onobject*", and "*offobject*" respectively.

The algorithm to recognise boss(es), hole(s) and undercut(s) with constant diameter.

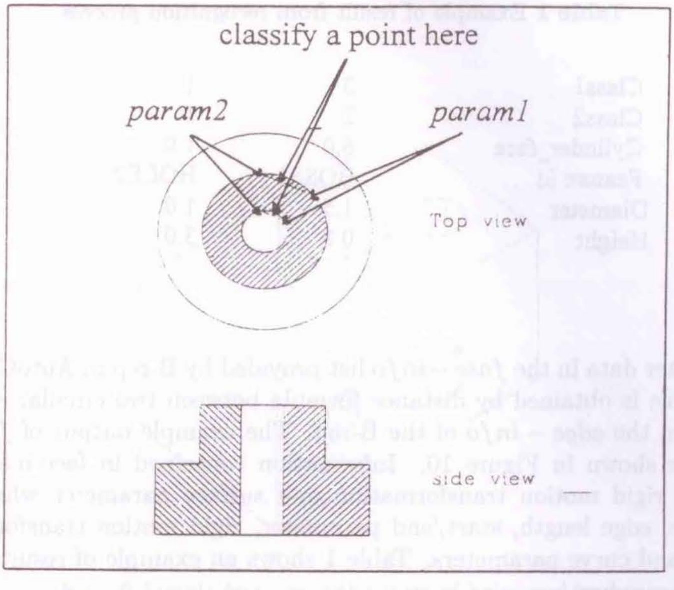


Fig. 4 Intersecting features: a hole inside a boss

(DEFUN RECOGNISE_HOLE_BOSS_UNDERCUT)
(find cyclface)
(foreach cylinder face *cyl_face* in *face_list*
  (find circular edges of *cyl_face*
    (foreach circular edge e_i
      (find the starting and ending parameter *param*1, *param*2
      (find the midpoint of *param*1, *param*2
      (find the point at 1/2 of the midpoint of *param*1, *param*2
      (find the point relative to solid
);foreach
  (IF point in *class*1 is 1 and in *class*2 is 1
    THEN identify through hole)
  (IF point in *class*1 is 2 and in *class*2 is 1
    THEN identify blind hole)
  (IF point either in *class*1 is 3 and *class*2 is 2 or in *class*1 is 2 and in *class*2 is 3
    THEN identify a boss)
  (IF point in *class*1 is 2 and in *class*2 is 2
    THEN identify undercut)
  (IF point both in *class*1 and in *class*2 is 0
    THEN identify "*invalid*" hole or boss)
);foreach

The classification point of *class*1 and *class*2 is that if the point is *offobject*, *onobject*, *inobject* or *failed*, the value is 1, 2 3 and 0 respectively.

Once the hole or boss has been identified, the next step is to determine the diameter or radius, and depth parameters of this feature. The diameter of the boss and hole is calculated

Table 1 Example of result from recognition process

Class1	3	1
Class2	2	1
Cylinder_face	6.0	1.0
Feature id	BOSS1	HOLE2
Diameter	1.5	1.0
Height	0.8	3.0

using the perimeter data in the $face-info$ list provided by B-rep in AutoCAD. The height of the boss or hole is obtained by distance formula between two circular edges which can be extracted from the *edge* $- info$ of the B-rep. The example output of $face - info$ and edge-info list are shown in Figure 10. Information contained in face-info list is surface type, perimeter, rigid motion transformation and surface parameter whilst in edge-info list is curve type, edge length, start/end parameter, rigid motion transformation matrix, start/end point and curve parameters. Table 1 shows an example of results obtained from the recognition procedure.

As the features (hole or slot) are identified, the associated information is stored in *feature_list_data* structure (see Figure 8 or Figure 9). The system can also detect invalid holes.

```
"file name C:/acadwin/napwork/holes.dat"
"facelist"
(1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0)

"edgelist"
(14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0
30.0 31.0 32.0 33.0)

"Face2edge_list"
1.0 (15.0 14.0)
2.0 (17.0 16.0)
3.0 (17.0)
4.0 (19.0 18.0)
5.0 (18.0)
6.0 (21.0 20.0)
7.0 (20.0)
8.0 (25.0 24.0 23.0 22.0)
9.0 (28.0 27.0 26.0 22.0 19.0 14.0)
10.0 (31.0 30.0 29.0 27.0)
11.0 (33.0 32.0 30.0 24.0 15.0)
12.0 (33.0 31.0 28.0 25.0)
13.0 (32.0 29.0 26.0 23.0 21.0 16.0)
```

Fig. 5 A sample output from B-rep data structure for Figure 6

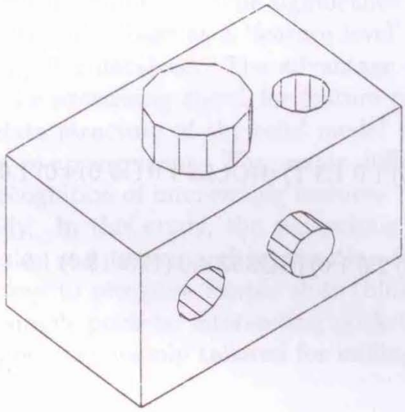
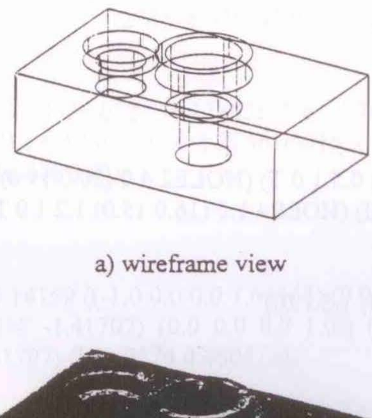
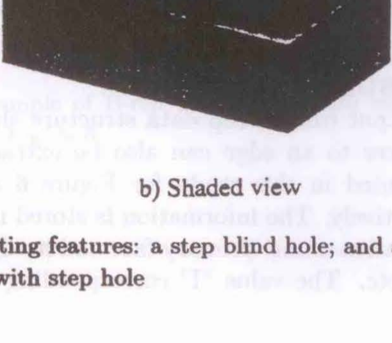


Fig. 6 Simple features: solid view of holes and bosses



a) wireframe view



b) Shaded view

Fig. 7 Interesting features: a step blind hole; and a boss and hole ('protrusion') with step hole

```
"cylinder_face_list"
(6.0 4.0 2.0 1.0)

"Hole_list"
((HOLE1 2.0 (17.0 16.0) 1.0 1.5 T) (HOLE2 1.0 (15.014.0) 1.0 3.0 nil))

"Boss_list"
((BOSS1 6.0 (21.0 20.0) 2.0 1.0) (BOSS2 4.0 (19.0 18.0) 1.0 1.0))

"Undercut_list"
nil
```

Fig. 8 Recognition result (feature_list_data structure) for simple feature of Figure 6

```
"cylinder_face_list"
(13.0 6.0 4.0 3.0 1.0)

"Holes_list"
((HOLE1 6.0 (22.0 21.0) 0.5 1.0 T) (HOLE2 4.0 (20.0 19.0) 1.0 1.0 T) (HOLE3
3.0 (18.0 17.0) 1.0 2.0 nil) (HOLE4 1.0 (16.0 15.0) 1.2 1.0 T))

"Boss_list"
((BOSS1 13.0 (36.0 34.0) 1.5 1.0))

"Undercut_list"
nil
```

Fig. 9 Recognition results for intersecting feature of Figure 7

5 EXPERIMENTAL RESULTS

Figure 5 shows a sample output from B-rep data structure shown in Figure 6. The edge coordinates and adjacent faces to an edge can also be extracted. The results from the recognition system implemented in this study for Figure 6 and Figure 7 are shown in Figure 8 and Figure 9 respectively. The information is stored in feature_list_data structure consisting of feature identification name, primary face and it edges, feature parameters such as diameter, length, height etc. The value 'T' corresponding to a blind hole otherwise a through hole.

6 DISCUSSION AND CONCLUSION

Features represented in B-rep file can be identified and extracted algorithmically. The system has demonstrated a possible approach to recognise feature and extract feature information such as diameter, length and height. The significance of this approach is the ability to communicate with geometry database at a 'feature level' and capture information not explicitly stored in the computer database. The advantage of the boundary point classification algorithm is that the processing speed for feature recognition is faster compared to graph-based since the data structure of the solid model is input directly to the recognition system without any re-arrangement. The major difficulty common to all different approaches has been the recognition of intersecting features. When features intersect, their topology changes drastically. In this study, the recognition system currently can handle simple features such as holes and bosses and intersecting features such as step holes or bosses. Research is underway to recognise simple slots (blind, through, corner slots) and intersecting slot; pockets (simple pockets, intersecting pockets, pocket with different 'wall' height and angle, n-sided pockets), mainly tailored for milling processes.

```
"faceinfo"
1.0
(1 5.78774 ((-1.0 0.0 0.0 7.69368) (0.0 1.22461e-016 -1.0 6.81603) (0.0 -1.0 -
1.22461e-016 -1.41707) (0.0 0.0 0.0 1.0)) (0.460574 0.460574))
2.0
(1 5.27711 ((1.0 -7.4983e-032 6.12303e-016 4.74686) (1.12791e-048 -1.0 -1.22461e-
016 5.63862) (6.12303e-016 1.22461e-016 -1.0 5.5929e-016) (0.0 0.0 0.0 1.0))
(0.419939 0.419939))
3.0
(0 2.63855 ((-1.0 -7.4983e-032 -6.12303e-016 4.74686) (-1.12791e-048 -1.0
1.22461e-016 5.63862) (-6.12303e-016 1.22461e-016 1.0 -1.0) (0.0 0.0 0.0 1.0)) ((-
6.12303e-016 1.22461e-016 1.0)))

"Edgeinfo"
14.0
(1 2.89387 -3.14159 3.14159 ((-1.0 0.0 0.0 7.69368) (0.0 1.22461e-016 -1.0 6.81603)
(0.0 -1.0 -1.22461e-016 -1.41707) (0.0 0.0 0.0 1.0)) (8.15426 6.81603 -1.41707)
(8.15426 6.81603 -1.41707) (0.460574 0.460574))
15.0
(1 2.89387 -3.14159 3.14159 ((-1.0 0.0 0.0 7.69368) (0.0 1.22461e-016 -1.0 3.82957)
(0.0 -1.0 -1.22461e-016 -1.41707) (0.0 0.0 0.0 1.0)) (8.15426 3.82957 -1.41707)
(8.15426 3.82957 -1.41707) (0.460574 0.460574))
```

Fig. 10 Example of B-rep output (face-info and edge-info) for component in Figure 6

REFERENCES

[1] B. K. Choi et al, *Automatic Recognition of Machined Surfaces from a 3D Solid Model*, Computer Aided Design **16, No. 2** (1984), 81–86.

[2] J. Corney & D.E.R. Clark, *Method for finding Holes and Pockets that Connect Multiple Faces in 21/2D object*, Computer Aided Design **23, No.10** (1991), 658–668.

[3] L. De Floriani, *Feature Extraction from Boundary Models of Three Diamensional Objects*, IEEE Trans. Pattern Anall. and Mach Intell **11, no.8** (1989), 785–798.

[4] J.C.E. Ferreira & S Hinduja, *Convex-Hull-Based Feature Recognition Method For 2.5D Components*, Computer Aided Design **22, No. 1** (1990), 41–49.

[5] M.C. Fields & D.C. Anderson, *Fast Feature Extraction For Machining Applications*, Computer Aided Design **26, No. 11** (1994), 803–812.

[6] M.R. Henderson et al., *Boundary Representation-based Feature Recognition*, Advances in Feature Based Manufacturing. Editor: Shah J J et al, Elsevier, 1994, pp. 15–38.

[7] M R Henderson & D Anderson, *Computer Recognition and Extraction of Form Features: A CAD/ CAM Link*, Computer in Industry, 5 (1984), 329–339.

[8] S. Joshi & T.C. Chang, *Graph-Based Heuristics for Recognition of Machined Features from a 3D Solid Model*, Computer Aided Design, Vol.20, No.2 (1988), 58–66.

[9] Y.S. Kim, *Volumetric Feature Recognition Using Convex Decomposition*, Advances in Feature Based Manufacturing. Editor: Shah J J et al., Elsevier, 1994, pp. 39–63.

[10] V.S. Kulkarni, *A System for Automatic Extraction of 3D Part Features Using Syntactic Pattern Recognition Techniques*, International Journal Production Research **33, No. 6** (1995), 1569–1586.

[11] T. Laakko and M. Mantyla, *Feature Modelling By Incremental Feature Recognition*, Computer Aided Design **25, No. 8** (1993), 479–492.

[12] S. Prabhakar & M.R. Henderson, *Automatic Form-feature Recognition using Neural-Network-based Techniques on Boundary Representations of Solid Models*, Computer Aided Design **24, No.7** (1992).

[13] L.C. Sheu & T.L. James, *Representing Scheme for defining and Operating Form Features*, Computer Aided Design . **25, No.6** (1993), 333–346.

[14] S.M. Staley et al, *Using syntactic Pattern Recognition to Extract Feature Information from a Solid Geometric Data Base*, Computers in Mechanical Engineering (1983), 61–66.

[15] Y.J. Tseng & S.B. Joshi, *Recognising Multiple Interpretations of Interacting Machining Features*, Computer Aided Design **.26, No. 9** (1994).

[16] J. Vanderbrande & A. Requicha, *Geometric Computational For The Recognition of Spatially Interacting Machining Features*, in Advances in Feature Based Manufacturing. Editor: J.J. Shah et al, Elsevier, 1994, pp. 83-105.