Pertanika J. Sci. & Technol. 5(2): 231-240 (1997)

ISSN: 0128-7680

Recognition of Machined Features from Solid Database of Prismatic Components

Napsiah bt Ismail and Nooh bin Abu Bakar
Business and Advanced Technology Centre (BATC)
Universiti Teknologi Malaysia
54100 Kuala Lumpur, Malaysia

Received 5 August 1996

ABSTRAK

Pengautomatan perancangan proses memerlukan pengecamam sifat secara langsung daripada sistem CAD. Kertas in membentangkan satu kaedah baru dalam pengecaman sifat yang dimesin dengan menggunakan konsep pengkelasan titik sempadan serta berasaskan teknik logik. Pemodelan bersepadu berasaskan perwakilan sempadan telah digunakan bagi memodelkan komponen prismatik. Sistem ini telah dibangunkan dalam persekitaran AutoCAD dan bahasa AutoLISP telah digunakan dalam pembangunan sistem pengecaman kerana ianya membolehkan capaian terus kepada pengkalan data. Keputusan ujikaji dibentangkan untuk menunjukkan keupayaan algoritma pengecaman sifat ini. Kertaskerja ini memberi penumpuan terhadap sifat yang dimesin dari jenis penurunan dan penaikan.

ABSTRACT

The automation of process planning requires features to be recognized directly from a computer aided design (CAD) system. This paper presents a new technique for recognition of machined features using point classification technique with a logic-based approach. Boundary representation of solid modelling is used to model a prismatic component. The system is developed entirely in the AutoCAD environment, and the AutoLISP language was used to build the recognition system as it has direct access to the database. Test results are presented to demonstrate the capabilities of the feature recognition algorithm. This paper concentrates on depression and protrusion type machined features.

Keywords: computer aided design, machined features, feature recognition, process planning, point classification

INTRODUCTION

Automatic feature recognition systems are likely to be an essential requirement for bridging the gap between computer aided design (CAD) databases and automation of process planning (PP) function. CAD is used to assist the engineer during the design process and create geometrical databases of the part to be manufactured. Part description in CAD models is in the form of basic geometry and topology that is unsuitable for direct application to process planning. A PP system generates process plans from the engineering specifications of the finished part. Today, most PP systems are running as stand alone systems. In order to automate the process planning activity, engineering specifications must be available directly from the CAD database. The use of high-level part representation in the form of features such as holes, grooves and pockets (see *Fig. 1*) is generally regarded as more suitable for the process planning reasoning tasks. Therefore, the automated process planning systems (APPS) must have the ability to recognize the features of a part automatically so that the machining operations and other machining data associated with the identified feature can be determined and stored in the database which forms a useful feature model for the generation of process plans.

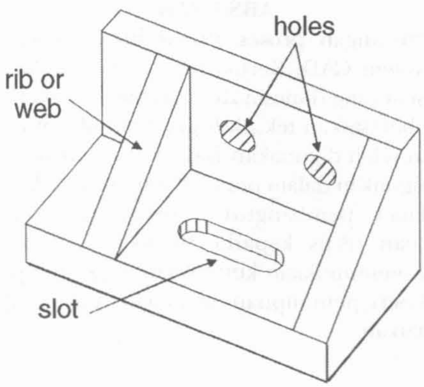

Fig. 1. A part described in terms of features

The most common methods used to represent objects in CAD systems are wireframe, surface modelling and solid modelling (Zeid 1991). Wireframe modelling represents objects by lines and points (vertices). A surface model of an object is a more complete and less ambiguous representation than the wireframe model. It defines an object by surfaces, edges and vertices, but does not store the topology (connectivity) of these entities. Solid modelling, such as boundary representation (B-rep) and constructive solid geometry (CSG), provides more complete, valid and unambiguous representation than surface modelling. In this work, B-rep solid modelling is used to describe the part. A B-rep solid model is constructed using entities like points (or vertices), lines (or edges) and faces (see *Fig. 2*). A point, or vertex, is a zero-dimensional entity defining a point in 3D-space which can be described as a triplet of numbers: (x, y, z). A line is a one-dimensional entity which can be drawn along one straight or curved axis. A line can be defined in several forms, depending on its types: straight, circular arc, ellipse, parabola, hyperbola, helix, or general nth-order curve. A surface, or face, is a two-dimensional entity because it can be drawn in a plane. It can be in the form of a plane, cylinder, sphere, cone, or as nth-order surface.

Many researchers have focused their attention on the problem of automated feature recognition. The B-rep based feature recognition approaches have employed various techniques such as syntactic pattern recognition, graph-

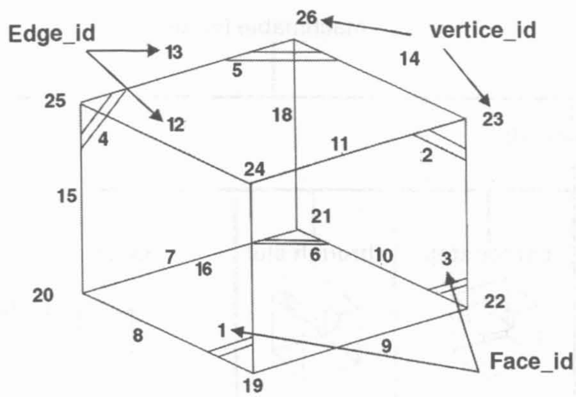

Fig. 2. B-rep labelled model

based, expert system and logic-based, neural network and hybrid approach (Shah 1991; Henderson *et al.* 1994). Joshi and Chang (1988) have proposed the attributes adjacency graph (AAG) to recognize polyhedral features. The main limitation of their approach is that it cannot detect features with non-planar faces, thus excluding features like cylindrical holes or bosses. Prabhakar and Henderson (1992) used a neural net technique for feature recognition. Only simple features can be recognized with no intersecting features. This paper presents another technique for feature recognition using a point classification approach with logic-base to solve some of the limitations found in the other approaches.

DEFINITION OF FEATURES

A feature may be generally described as a set of geometric entities (faces, edges) which together define a topology and geometry (Wang 1993). In this work, feature is defined as the grouping of a set of faces on the surface of the part that correspond to shapes that are commonly found on machined parts which can be grouped as *depression* and *protrusion* type features. Depression features are produced by machining away the material comprising the feature volume. Protrusion features such as boss or block are less simple from the machining point of view, since material to be machined away does not correspond to the feature volume, but rather that which surrounds it. The features in *Fig. 3* are considered in this work.

Intersecting features are also recognized. As an example, consider a prismatic part in *Fig. 4.* Notice that slot1 intersects with slot2 because of one of the entrance faces to slot2 lies in the side face of slot1.

FEATURE RECOGNITION SYSTEM

The feature recognition system is developed using both the logic-based approach and point classification concept. The AutoCAD Advanced Modelling Extension

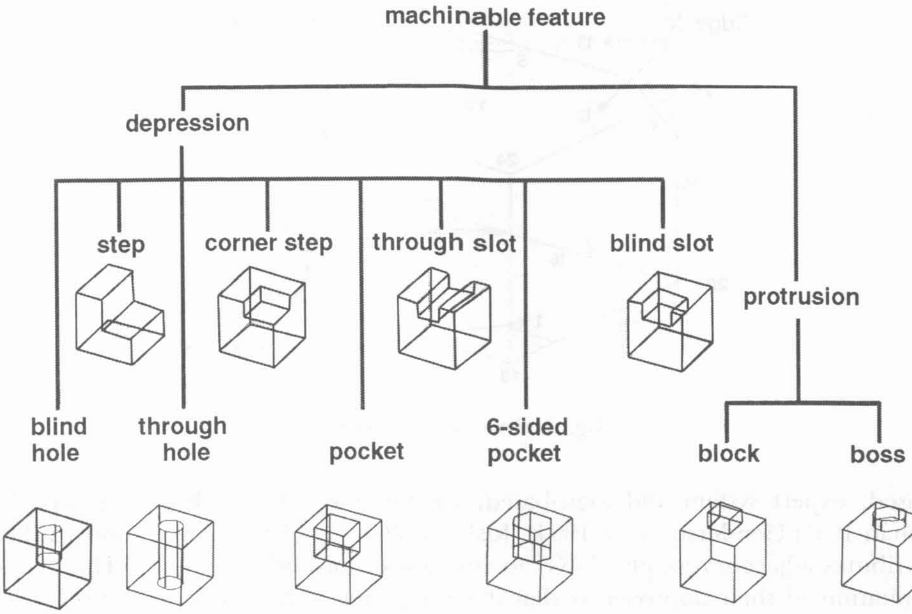

Fig. 3. Machinable features hierarchy considered in this work

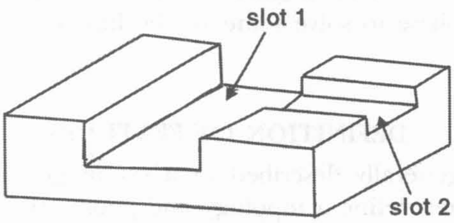

Fig. 4. Intersecting features

(AME 1993) module is used to create the B-rep solid models. A software module was developed to extract the *entity_name* list of entities that represent the solid model and the environment setting of AutoCAD. The *entity_name* of the main block for the composite solid model currently being displayed (i.e. in the CURRENT LAYER) is then identified and retrieved for further processing in the recognition system. The main block is a group of entities (i.e. lines, points etc.) that AutoCAD treats as a single entity in the current layer. The previous entities that made up the composite solid model are secondary entities and are demoted to the AME_FRZ LAYER (frozen layer). The next module concentrates on the identifying of faces, edges and vertices for point classification purposes using AME's interface operation functions.

Point Classification Approach to Feature Recognition

The concept of point classification is the concept that points in space can be classified relative to the object in solid modelling, as either lying *inside* the object, *outside* the object or *on* the object. The classification points of an *edge_loop* are shown in *Fig. 5*. The values 1, 2, 3 and 0 are respectively assigned if the point is *offobject, onobject, inobject* or *failed.* Two sets of classification points are required for each *edge_loop*, one set of points lying inside the *edge_loop* and another set outside the *edge_loop*. This point classification obtained is then applied in a logic-based programming technique to recognize cylindrical face features and also polyhedral features.

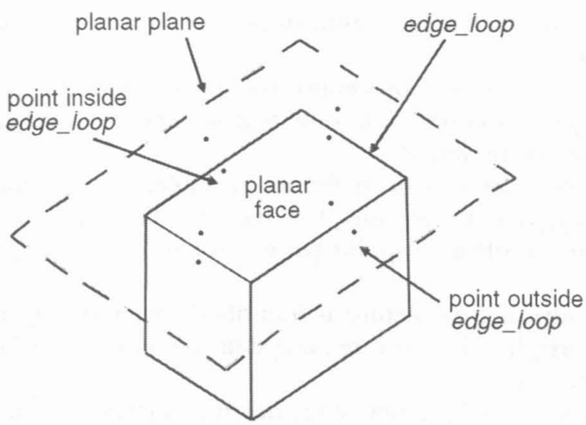

Fig. 5. Determination of point at edge_loop

Logic-based Programming Approach

The logic-based approach uses artificial intelligence techniques to develop a set of feature rules. The algorithms identify a feature based on certain pre-specified rules that are characteristic to the feature. The following set of heuristic rules is used to describe a through hole feature:

IF circular edges are found and

point in *cir_edge1* is 1 and *cir_edge2* is 1

THEN identify *through hole*

Cylindrical Face Features

Cylindrical features can be recognized from a B-rep model from the existence of cylindrical faces and their associated circular edges (see *Fig. 6*). The feature recognition algorithm searches for holes, bosses or undercuts by searching for a set of cylindrical face, entrance face and bottom face. The cylindrical face is determined first, then its associated edges are examined using the point

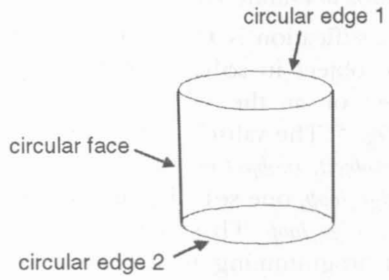

Fig. 6. B-rep model of a cylinder

classification approach. The algorithm stops when a pattern is identified for a particular feature.

Tables 1 and 2 show the procedure for determining the cylindrical faces and circular edges (for entrance face or bottom face) respectively for features such as hole, boss or undercut.

After a set of faces has been found for hole, boss or undercut, point classification is applied at every *edge_loop* (i.e. circular edge) to determine the feature type. The classification point pattern for features recognized is shown in Table 3.

Once a cylindrical type feature is identified, the resulting information is stored in a feature_data list. The feature_data list (see Table 5) contains the following information:

(feature id, face id, edges, diameter, height, sub-type of feature)

Each element of this list is described below:

- feature id; a unique feature identification e.g. Hole1, Hole2, Boss1, Undercut1
- face id; the cylindrical face id of the feature
- edges; the list of associates edges for face id

TABLE 1

Procedure for determining cylindrical faces

```
(Defun FIND_CYLFACE (/faceid facesinfo)
    (determine faceid face_list)
    (get facesinfo)
    (foreach faceid in facesinfo
        (if
            (= (nth 0 facesinfo) 1); cylindrical face
            (append cylinderface_list))
        (else
            (append non_cylinderface_list))
    ); foreach
);defun
```

TABLE 2

Procedure for determining circular edge

```
(Defun FIND_CIREDGE (/ edgeid edgeinfo)
    (determine edgeid edge_list)
        (get edgeinfo)
        (foreach edgeid edgeinfo
            (if ( = ( nth 0 edgeinfo) 1) ; circular edge
                (append ciredge_list)
            (else
                (append non-circular_list))
    ); foreach
); defun
```

TABLE 3

Feature types recognized

	Point classification		Feature recognised
	cir_edgel	cir_edge2	
	1	1	Through hole
Point value	2	1	Blind hole
	1	2	Blind hole
	3	2	Boss
	2	3	Boss
	2	2	Undercut/ Internal slot

- diameter; diameter of feature
- height; height of feature.
- sub-type of feature; A flag to denote whether the feature is a through hole or a blind hole. 'T' denotes through feature and 'NIL' denotes blind feature.

Polyhedral Features

Polyhedral features such as pockets, slots and steps can also be recognized using point classification techniques with logic-based. As an example, point classification patterns for *edge_loop* of the through slot bottom face, F1 (see *Fig. 7*) are in Table 4. Either pattern can identify the bottom face of the through slot. Similar procedures are used for other feature types (depression and protrusion features), i.e. pocket, n_sided pocket and protrusion block.

As a through slot feature is identified, the resulting information is stored in a through_slot_list (see Table 6) of feature_data. The list contains the following information:

(feature id, bottom_face id, bottom_edges, wall faces)

Each element of this list is described below:

- feature id; a unique feature identification e.g. Thruslot1, Thruslot2
- bottom_face id; the bottom face id of the feature
- bottom_edges; the list of associates edges for bottom_face id
- wall faces; the list of faces adjacent to bottom_face id

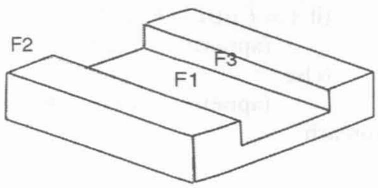

Fig. 7. Through slot

TABLE 4
Point classification of edge-loop FI of through slot

Point calssification of edge_loop FI	Edges	e_1	e_2	e_3	e_4	Feature recognized
inside boundary		2	2	2	2	through
outside boundary	pattern 1	1	3	1	3	slot
	pattern 2	3	1	3	1	

RESULTS AND DISCUSSION

The system was tested on some typical engineering components. The recognition system has the capability to recognize simple features and intersecting features.The part tested is depicted in *Fig. 8.* The system identifies the part has twenty-five faces, sixty edges and recognizes seven features.

Table 5 shows the results obtained from the recognition system for cylindrical face features of *Fig. 8.* The recognized features are one blind hole (HOLE1), one through hole (HOLE2), and one boss (BOSS1). As an example, the result "(HOLE1 4.0 (31.0 30.0) 0.6 1.25 nil)" can be read as follows:

HOLE1 comprises *bottom_face,* where face_id is number 4.0 with its associated edges, *circular edge-1* and *circular edge-2* denote as edge_id, 31.0 and 30.0 respectively, diameter of hole is 0.6 unit, height 1.25 unit, and the feature is a blind hole.

As seen from Table 6, two pockets, one slot and one step have been recognized for polyhedral features of *Fig. 8.* The pocket_list contains the following information:

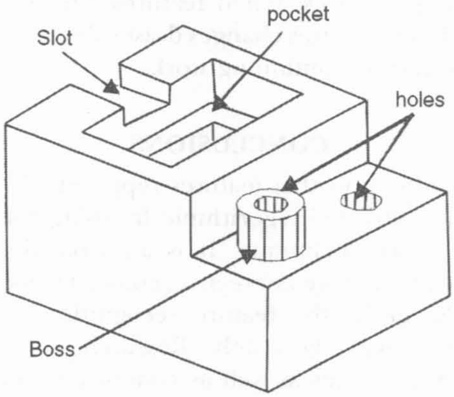

Fig. 8. Prismatic part

(feature id, bottom_face id, bottom_edges, entrance face id, entrance edges, face_loop, wall faces)

and for the step_list contains information such as:

(feature id, bottom_face id, bottom_edges, wall faces)

The POCKET1's entrance face and its associated edges can not be determined because POCKET1 and THRUSLOT1 are intersecting features in which THRUSLOT1 has disturbed the edge topology of POCKET1. This is the major difficulty common to all different approaches of feature recognition systems in

TABLE 5
The result for cylindrical face features for *Fig. 8*

"Holes_list"
((HOLE1 4.0 (31.0 30.0) 0.6 1.25 nil) (HOLE2 1.0 (27.0 26.0) 0.5 1.25 T))
"Boss_list"
((BOSS1 3.0 (29.0 28.0) 1.0 0.5))

TABLE 6
The result for polyhedral features for *Fig. 8*

Pocket_list
((POCKET1 13.0 (57.0 56.0 54.0 55.0) "not available" "not available" (16.0 17.0 14.0 15.0))
(POCKET2 5.0 (35.0 34.0 33.0 32.0) 13.0 (43.0 42.0 39.0 37.0) (7.0 8.0 9.0 6.0)))

Slot_list
((THRUSLOT1 10.0 (47.0 45.0 46.0 44.0)(12.0 11.0)))

Step_list
((STEP1 10 (70.0 69.0 68.0 67.0) 19.0)))

recognizing intersecting features. When features intersect, the face and edge pattern associated with the features changes drastically. This problem is presently being looked into as part as continuing work.

CONCLUSIONS

The experimental results show that features represented in B-rep solid model can be identified and extracted algorithmically using the point classification approach with logic_based technique. It is anticipated that the system can ultimately function as an interface between a geometric modeller and automated process planning. Currently, the feature recognition system can recognize features that have concave edges only. Research is ongoing to recognize features that have convex edges as well as complex features which have both planar face and cylindrical face adjacent to each other.

REFERENCES

AutoCAD Advanced Modelling Extension (AME). 1993. Release 2.1.

HENDERSON, M.R., G. SRINATH, R. STAGE, K. WALKER and W. REGLI. 1994. Boundary representation-based feature identification. In *Advances in Feature-based Manufacturing*, ed. J.J. Shah, M. Mantyla and D.S. Nau, p.15-38. Elsevier Science.

JOSHI, S. and T.C. CHANG. 1988. Graph-based heuristics for recognition of machined features from a 3d solid model. *Computer Aided Design* **20(2)**: 58-66.

PRABHAKAR, S. and M.R. HENDERSON. 1992. Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models. *Computer Aided Design* **24(7)**:

SHAH, J.J. 1991. Assessment of feature technology. *Computer Aided Design* **23**: 331-343.

WANG, M.T. 1993. Manufacturing feature extraction and machined volume decomposition in the computer-integrated feature-based design and manufacturing planning environment. *Computers in Industry* **23**: 75-86.

ZEID, I. 1991. *CAD/CAM Theory and Practice.* McGraw-Hill.