Journal of Intelligent Manufacturing (1995) **6**, 155–162

A graph-based expert system approach to geometric feature recognition

P.K. VENUVINOD and S.Y. WONG

Department of Manufacturing Engineering, City Polytechnic of Hong Kong, Hong Kong

Received September 1993 and accepted May 1994

Geometric feature recognition is a crucial task in the development of concurrent engineering software. This paper presents a new methodology for geometric feature recognition which combines the advantages of face-edge adjacency graphs and expert systems. The methodology uses several new concepts such as enhanced winged edge data structure (EWEDS) and multi-attributed adjacency graphs (MAAG). The object model is presented as a set of facts. The rules for the recognition of each feature are derived from the corresponding feature-MAAG. This simplifies the process of writing the rules while enabling the inclusion of new features into the rule base as they are encountered.

Keywords: Feature recognition, adjacency graphs, expert systems

1. Introduction

Concurrent engineering is now widely recognized as a major answer to the ever increasing demand in the world market for the reduction of lead times involved in the design and manufacture of products. Current industrial practices concerning the design–manufacture cycle are iterative wherein the designs move back and forth between designers and manufacturing engineers. This is because designers cannot always foresee the problems that might arise downstream in the manufacturing phase. Concurrent engineering aims to overlap the design and manufacturing phases as much as possible so that the total lead time is significantly reduced. Designers these days typically make use of CAD systems which merely make the design process more convenient while leaving unaddressed the evaluation of the designs in terms of manufacturing and assembly lead times and costs. Therefore, for concurrent engineering to succeed, there is a need to develop computerized design for manufacture (DFM) and design for assembly (DFA) tools which facilitate the integration of manufacturing and assembly criteria into the product design process.

Figure 1 outlines the broad strategy adopted at the City University of Hong Kong in the development of computer automated DFA (Venuvinod, 1993). It is possible to apply a similar strategy in developing computer automated DFM tools provided that the three phases devoted to 'Technological pre-processing', 'assembly pre-processing' and 'UMass DFA analysis' are replaced by

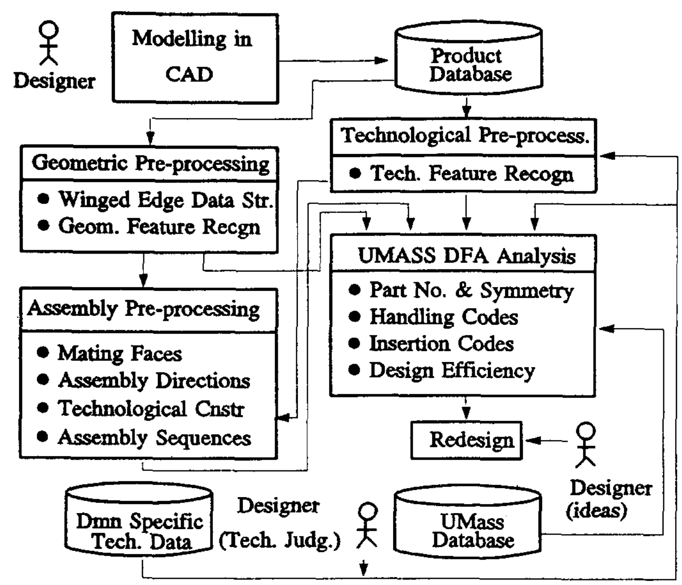

Fig. 1. A strategy for the automated analysis of assembly designs.

the corresponding DFM tasks. It may be noted that these phases are invariably domain-specific, i.e. they are dependent on the processes and/or the technological practices specific to the targeted industry or, even, company. In contrast, the phases devoted to product databases and geometric pre-processing tend to be generic in the sense that they are common to DFA and diverse DFM tasks.

This realization of the commonality amongst diverse DFM/DFA tasks is useful in enabling a common link to CAD databases and in sharing the same geometric preprocessing module(s) amongst the array of computerized tools necessary for facilitating concurrent engineering.

A frequently occurring task during the geometric preprocessing phase in most DFM/DFA exercises is the recognition of geometric features. A geometric feature is a descriptor of a subset of the geometric model of the object whose presence is relevant within the given functional context. For instance, in the context of computer-aided process planning (CAPP) of a component to be machined on a numerically controlled (NC) machine, the recognition of a 'pocket' feature may result in the decision to include an appropriate 'pocket milling' canned cycle in the NC program. Likewise, it could result in a different insertion code while applying the UMass system of DFA (Boothroyd and Dewhurst, 1983).

Thus, the problem of automating geometric feature recognition has received much attention in recent years. Amongst the major strategies adopted in this context are: (1) syntactic pattern recognition; (2) application of the theory of automata; (3) decomposition of the object into layers; (4) graph-based techniques; and (5) the utilization of rule-based expert systems.

However, with the exception of Henderson and Anderson's methodology (Henderson, 1984; Henderson and Anderson, 1984), which belongs to strategy (5), most of these approaches have been able to recognize only a limited set of polyhedral and/or cylindrical features as a consequence of the strategies utilized in feature representation, definition and inference. (It may be noted that the problem of recognizing features characterized by sculptured and other non-analytical surfaces has not been addressed so far.) In Henderson (1984), each feature of interest is defined by using a set of rules of logic implemented in PROLOG. An object is represented as a set of facts and these facts are searched to satisfy the feature rules. Since any definable topological or geometrical property can be used in a feature rule, in principle, this approach is unlimited in scope.

One problem with Henderson and Anderson's expert system-based approach needs to be noted. This pertains to the absence of a clear and systematic procedure to guide the specification of the logic rule to extract a given feature although it is stipulated that rules for features must be 'written by determining the necessary and sufficient conditions for the feature of choice and expressing them in logic statement' (Henderson and Anderson, 1984). However, these necessary and sufficient conditions are expected to be specified on the basis of an intuitive understanding of the geometry of the feature. No systematic tools are made available for identifying the specific geometric characteristics that need to be incorporated in the logic rule. Consequently, this approach often leads to situations where the specification of logic rules itself becomes an *expert* task.

A review of the various approaches adopted so far for feature recognition suggests that the information required in specifying an unambiguous logic rule for a feature can be classified into three levels. At the highest level, one is concerned with the *topology* of the feature, i.e. with information pertaining to the adjacencies amongst the faces (F), edges (E), and vertices (V) composing the feature. In general, the information required at this level can be succinctly captured by a face–edge (FE) graph in which each node represents a labelled face and each arc joining a pair of nodes represents the fact that these faces have a common edge in the object. At the next level, one is concerned with the *coarse geometry* of the feature. Amongst the feature attributes which may be encoded at this level are whether a given pair of faces intersect at an edge in a concave (i.e. the angle, θ, between the faces when measured on the material side of the object is greater than 180°) or convex manner, and whether a given face is plane, curved, etc. In fact, there is considerable empirical evidence that human image understanding primarily relies on parsing of objects at deep concavities (Biederman, 1985). This suggests that the use of information concerning the concavity or convexity of edges is a powerful means for feature recognition. At the next level, one needs to consider *fine geometry* information concerning the angular orientations, dimensions, tolerances, etc. of the faces. However, a review of literature suggests that most features of practical interest are definable on the basis of information at the first two levels, i.e. topology and coarse geometry.

Joshi and Chang (1988) have developed an elegant graph-based approach to capture the topological information and some of the coarse geometry information. In particular, they utilize an attributed adjacency graph (AAG) which is an FE graph of the object with each arc carrying attribute 0 if its terminal nodes have a concave relation and 1 if they have a convex relation. The features in the object are subgraphs of the object-AAG and recognition of features involves searching for and identifying the subgraphs that correspond to a feature. They then use a heuristic method to identify components of the graph that could form a feature. The subgraphs thus separated are analysed with the help of a set of highly structured procedures, called the 'recognizer', utilizing IF–THEN clauses. However, in Joshi and Chang (1988), the implementation of this AAG-based approach was limited to the recognition of polyhedral (i.e. plane faced) features, although some suggestions were made for enabling the recognition of cylindrical features.

A problem associated with Joshi and Chang's approach is that the highly structured 'recognizer' procedure is likely to be too restrictive when new feature types need to be included in the rule base. A substantial restructuring of or the addition of supplementary procedures within the 'recognizer' will be needed to resolve the anomalies arising whenever new or unexpected features are encountered in the object. Thus, this procedure does not facilitate natural growth.

This paper describes a new feature recognition procedure that combines the advantage of unlimited scope for adding new features inherent in Henderson *et al.*'s expert system-based approach (Henderson, 1984; Henderson and Anderson, 1984) with the simplicity and rigour of Joshi and Chang's concept of AAG in capturing the essential topological and coarse geometry properties of a feature (Joshi and Chang, 1988). The new approach also incorporates some enhancements to the concept of AAG in order to enable the recognition of features beyond the domain of polyhedral features. Thus, the current implementation of the new approach has enabled the recognition of almost all the polyhedral and cylindrical features cited in previous literature.

2. CAD interface and the enhanced winged edge data structure

Consider now the first activity in the geometric preprocessing module in Fig. 1 which concerns the conversion of the object model created on a CAD system into a data structure suitable for geometric feature recognition. In the present work, it is assumed that the object model is available in boundary representation (B-Rep). The B-Rep approach is particularly attractive from the viewpoint of geometric feature recognition because features are usually characterized by the presence of a set of faces satisfying a given set of topological and geometric interrelationships. In B-Rep, unlike constructive solid geometry (CSG), such face information is explicitly available. Further, with the recent incorporation of B-Rep into IGES (Version 5.1), it is anticipated that B-Rep will increase in popularity (Mattei, 1993). AutoCAD Version 11, which supports surface operations, has been used as the CAD platform in the present work (in view of its popularity in Hong Kong and the authors' empirical observation that, for the same object, the DXF file output by AutoCAD is often significantly more compact than the corresponding IGES file).

The B-Rep object model however needs to be converted into a data structure appropriate for geometric feature recognition. Amongst the data structures used in the context of B-Rep, the winged edge data structure (WEDS) of Baumgart (Baumgart, 1974) is particularly attractive for the present purpose. WEDS is an edge-based data structure providing explicit information concerning the object's faces, edges and vertices. From each labelled face, there is a pointer to each of its boundary edges. Likewise, from each edge, there is a pointer to each of its bounding vertices (*vstart* and *vend*). Each edge occurs in exactly two faces, once in the clockwise orientation and once in the counter-clockwise orientation as viewed from the outside of the object. The two adjacent faces defining each edge are classified as clockwise (*fcw*) and counter-clockwise (*fccw*). The structure also includes information on next edge clockwise (*ncw*) and next edge counter-clockwise (*nccw*) in addition to previous edge clockwise (*pcw*) and previous edge counter-clockwise (*pccw*). Thus the relationships between adjacent faces, which are essential for feature recognition, are explicitly preserved.

However, Baumgart's WEDS in its original form (Baumgart, 1974) suffers from the following limitations:

(1) It is applicable only to polyhedral objects, i.e. it is not designed to describe cylindrical and other curved faces which are often encountered in engineering objects and in design for assembly;

(2) It does not explicitly contain the information concerning concavity or convexity of edges which, as discussed earlier, is essential for feature recognition;

(3) It does not explicitly contain the parametric information concerning the orientation of faces which is required for determining the concavity or convexity of edges.

An enhanced winged edge data structure (EWEDS) is now proposed with a view to overcoming the above limitations of WEDS. In order to facilitate the graph-based expert system approach to feature recognition (as described in the next section), the EWEDS is represented in terms of edge, vertex and face types of PROLOG unit clauses:

(1) *edge (edge_number, vstart, vend, fcw, fccw, ANGLE ncw, pcw, nccw, pccw);*

(2) *vertex (vertex_number, FIRST_EDGE_LIST, coordinates);*

(3) *face (face_number, first_edge, TYPE_OF_FACE, PARAMETRIC_DATA_LIST_LIST)*

where the fields in *UPPER CASE* are the enhancements added to the WEDS of Baumgart.

The enhancements are explained in the following:

(1) *TYPE_OF_FACE*—this is a label attached to each face to indicate its type. The present work has included only three types of surface—'*pl*' for a plane face bounded by a loop consisting of straight edges, '*dsc*' for a plane face bounded by a circular edge (i.e. a disc), '*cyl*' for a face lying on a cylinder. However, in principle, this concept can be extended to other surface types;

(2) *FIRST_EDGE_LIST*—this is included to replace *first_edge* in Baumgart's WEDS since, unlike the original

WEDS, coplanar and adjacent plane triangular surface patches created during object modelling have been merged into a single face using a *same face* operation (Wong, 1992). Thus it is possible that faces will have disjoint edge loops corresponding to lakes, holes or islands. Each of these loops requires a *first edge* for its identification. By including the *first_edge_list*, faces with multiple boundaries are explicitly specified.

(3) *PARAMETRIC_DATA_LIST_LIST*—each element of this list is in itself a list of parametric data concerning specific information about the face. For instance, the first element in the parametric_data-list_list *[[0, 0, −1, −1], [0, 0, 1]]* specifies that the face is lying in a plane described by equation $Ax + By + Cz + D = 0$ with $A = 0$, $B = 0$, $C = -1$, and $D = -1$ (x, y and z are the Cartesian variables with respect to the global coordinate system specified by the designer at the time of modelling the object on a CAD system). The second element, *[0, 0, 1]*, in the same list records the fact that the unit normal vector of the face (pointing away from the material side of the object) is $\{0\boldsymbol{i} + 0\boldsymbol{j} + 1\boldsymbol{k}\}$ where $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are the unit vectors in the directions of Cartesian axes x, y and z respectively. The contents of the list depends on the *type_of_face*. In general, the list [parametric_data_list_1, parametric_data_list_2,,, parametric_data_list_n] describes a face whose complete specification requires n parameters depending on the *type_of_face*.

(4) *ANGLE* (θ)—this is the angle on the material side of the object between the pair of faces intersecting at the edge when measured in a section normal to the edge. The angle is easily calculated from the unit normal vectors of the pair of faces radians. This angle is subsequently used in the feature recognition module to determine the concavity, convexity or otherwise of the edge.

All the information contained in the EWEDS is either explicitly or implicitly available in the DXF file output by AutoCAD (likewise, it is also available in the IGES 5.1 file). The authors have successfully developed and tested a software interface for DXF files which is capable of automatically generating the EWEDS for objects with plane and cylindrical faces. Future efforts are proposed to be directed towards developing interfaces with the capability for describing other types of curved faces as well. The recently developed IGES 5.1 should be a worthwhile target since it includes B-Rep.

3. Multi-attributed adjacency graph (MAAG)

The AAG (attributed adjacency graph) originally proposed by Joshi and Chang (1988):

(1) Does not have any node attributes, i.e. it cannot distinguish between different types of faces (plane, cylindrical, etc.); and

(2) Has only a limited number of arc attributes; in particular only two attributes are utilized—concave edges carry label '0' and convex edges carry '1'.

Thus, this form of AAG is capable of describing the topology and coarse geometry of only polyhedral objects.

With a view to extending the capabilities of the concept of AAG to more complex objects other than polyhedra, the concept of multi-attributed adjacency graph (MAAG) is now proposed. MAAG is derived by adding the following enhancements to AAG:

(1) Each node carries an attribute label: '*pl*', '*dsc*', '*cyl*', etc;

(2) The choice of arc attributes is extended according to the following scheme

(i) arc attribute *0* for a concave edge, i.e. $180° < \theta < 360°$,

(ii) arc attribute *1* for a convex edge, i.e. $0 < \theta < 180°$,

(iii) arc attribute *2* for an edge with $\theta = 360°$ (within a user-specified error band),

(iv) arc attribute *3* for a smooth edge, i.e. $\theta = 180°$, and

(v) arc attribute *4* for an edge with $\theta = 0$ (within a user-specified error band).

It may be noted that all the information necessary to determine the MAAG is explicitly available within EWEDS. Figures 2a and b respectively show an object and the corresponding MAAG. Note that the object includes three geometric features: a three-sided slot, a cylindrical projection and a quarter cylindrical projection in a corner pocket. The subgraphs corresponding to each of these features are highlighted in bold lines in Fig. 2b. These subgraphs will henceforth be called the feature-MAAGs.

4. An improved rule-based approach to feature recognition

It has already been noted in Section 2 that a feature is mainly characterized by the topological and geometric interrelationships existing between the faces making up the feature.

To develop this concept in some detail, consider the three-sided slot feature in the object shown in Fig. 2a. Clearly, the slot feature is made up of faces 3–7. However, it is intituively apparent that faces 4, 5 and 6 are the most important ones from the point of view of characterizing the feature as a three-sided slot. These may be called the *root faces* of the slot feature, and the subgraph within the feature-MAAG which is composed of the root

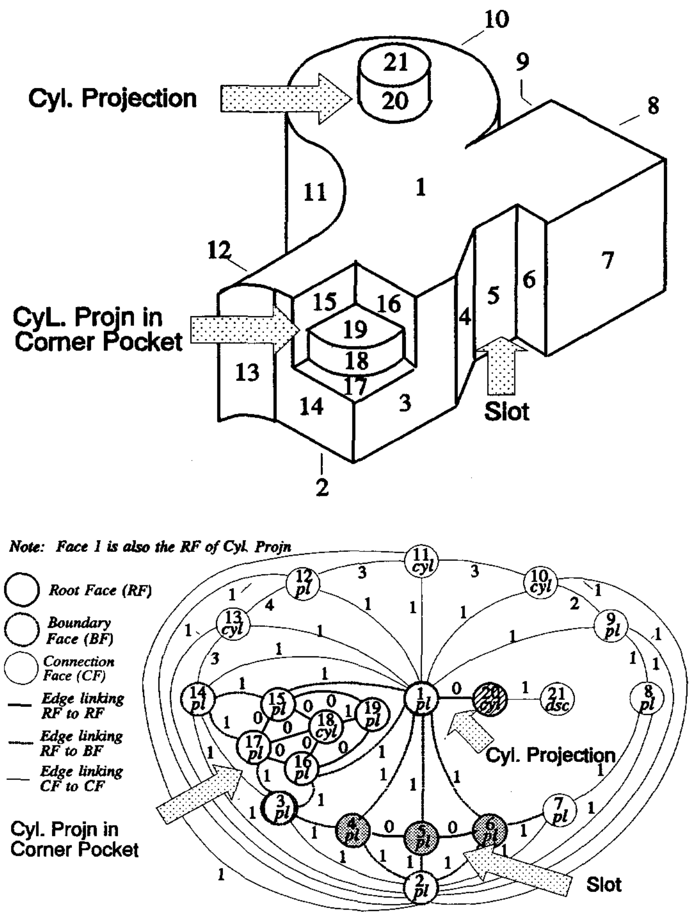

Fig. 2. An object and its multi-attributed adjacency graph (MAAG): (a) illustrating some types of edges, faces and features; (b) the MAAG of the object with the feature-MAAGs highlighted.

faces and the arcs linking pairs of root faces may be called the *root-feature-MAAG* of the slot feature. All features with an identical root-feature-MAAG belong to the family of three-sided slot features. The following heuristic definition is obtained on the basis of examining a large variety of geometric features: a root face of a feature is a feature face which has concave adjacency with at least one other feature face. Note further that an object face cannot simultaneously be a root face of two different features.

Feature faces which are not root faces may be called the *boundary* faces. Thus, faces 1, 2, 3 and 7 are the boundary faces of the slot feature in Fig. 2a. Boundary faces are not essential for recognizing the existence of a feature in a given object but are usually useful from the point of view of downstream processes such as process planning. For instance, while machining the slot feature, faces 1 or 2 are the faces from which the cutter enters and, therefore, may be labelled as the *entrance faces*. Likewise, faces 3 and 7 determine the direction in which the part with the female slot feature as the locating feature and the male counterpart would normally be assembled so that these may be labelled as the *top faces*. However sometimes, as in the case of a pocket to be milled, this distinction between entrance and top faces may not exist. Further, note that a boundary face of one feature can be a boundary or, even, a root face of another feature. For instance, face 1 in Fig. 2a is at once a boundary face of the slot feature, a boundary face of the cylindrical-projection-in-corner-pocket feature and a root face of the cylindrical-projection feature.

Boundary faces are determined by the interaction between the fine geometry of the root-feature and the rest of the object. Figure 3 illustrates this point. It is seen that a variety of three-sided slots each with a different topology, as illustrated in the first column in Fig. 3, is obtainable when the fine geometry (i.e. the dimensions and/or orientations of the root faces) is varied. Although all these slots belong to the general family of three-sided slots (since they have a common root-feature-MAAG), it is important to be able to discriminate between the different members of the family during the feature extraction and recognition phases. Further, it might sometimes be necessary to discriminate between two features from a functional point of view although they have the same topology and coarse geometry. The slots shown in the first row in Fig. 3 illustrate this point. For instance, the slots in the third and fourth columns of the first row would require significantly different process plans since the latter is a dovetail slot (i.e. the cutter cannot enter the slot from the top) whereas the former is not. Note that in writing the recognition rules for such functionally discriminated features it becomes necessary to include

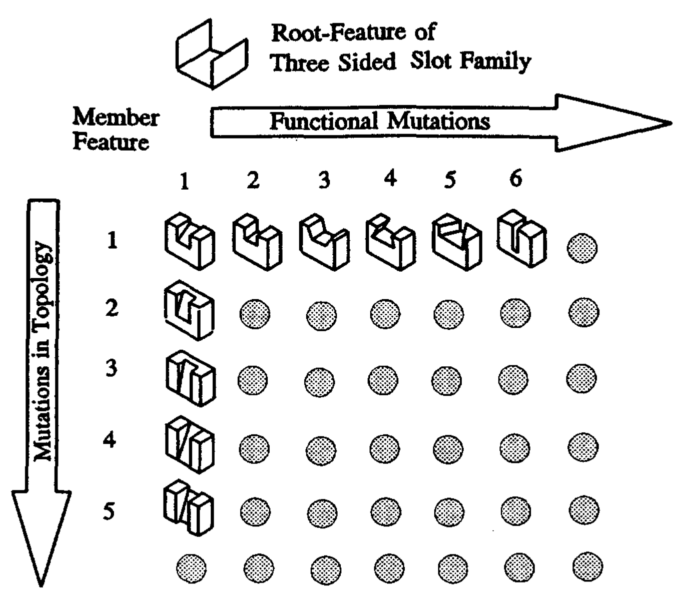

Fig. 3. A feature family (three-sided slots).

some fine geometry information too. However, the EWEDS has already been structured to accommodate the required fine geometry information. For instance, the angle between the root faces is explicitly available from the field labelled *angle* in the PROLOG clause for each root face. This problem is therefore easily solved by including the necessary angle conditions in the PROLOG rule for recognizing the desired feature.

Object faces that do not belong to any feature may be called the *connection faces*, i.e. from a feature recognition point of view, they simply exist to connect different features.

It follows from the above that it is highly useful to consider both root and boundary faces while recognizing features. However, a review of the literature on feature recognition indicates that most previous workers, with the exception of Henderson and Anderson (Henderson, 1984; Henderson and Anderson, 1984), have mainly utilized the root faces while performing feature recognition. In particular, Joshi and Chang perform feature recognition solely on the basis of root faces (Joshi and Chang, 1988). Further, they have developed an algorithmic procedure for extracting root-features from an object before proceeding to recognize each of them. Their root-feature extraction algorithm is based on the heuristic that 'a face that is adjacent to all its neighbouring faces with a convex angle does not form part of a feature'.

It is only recently that a solution has been found to the problem of algorithmically extracting features while including both root and boundary faces during the extraction process. Venuvinod and Yuen (1994) have developed an algorithmic procedure for partitioning an object-MAAG into subgraphs (called the feature-MAAGs) where each subgraph corresponds to a specific geometric feature present in the object and includes the face nodes for the feature and the arcs representing root face to root face adjacencies as well as the root face to boundary face adjacencies. The procedure however ignores boundary face to boundary face adjacencies since these are not considered crucial to geometric feature recognition.

Once the features have thus been extracted, all that remains is to recognize the extracted features one by one with the help of a rule base (we assume that the rule base has included the rules necessary for recognizing *all* the features of interest appearing in the object). Thus, the rule based expert system proposed in this paper differs from Henderson and Anderson's expert system approach (Henderson, 1984; Henderson and Anderson, 1984) where the entire object-MAAG had to be traversed to test for a match with each recognition rule in the rule base. This significantly reduces the computational complexity involved during the feature recognition phase and partially overcomes the major objection to the expert systems approach—that it is inherently expensive from the computational viewpoint (Joshi and Chang, 1988).

It follows from the above that a geometric feature is characterized by:

(1) The types (plane, cylindrical, disc, etc.) of the root and boundary faces making up the feature;

(2) The nature of the adjacency relationship (concave, convex, etc.) between each pair of root faces; and

(3) The nature of the adjacency relationship between each pair of root and boundary faces.

Note that all the above information characterizing a feature is contained within the feature-MAAG extracted from the object-MAAG by Venuvinod and Yuen's method (Venuvinod and Yuen, 1994). To illustrate this point, consider the subgraph in Fig. 2b which corresponds to the three-sided slot feature present in the object shown in Fig. 2a. The required information concerning the number and types of root faces and the adjacency relationship between each pair of root faces is explicitly available within the feature-MAAG of the slot. Thus, it is a straightforward task to write the following PROLOG rule for recognizing the slot's root face properties:

```
root_feat(slot,[A,B,C]):-
/* [side face 1, bottom face, side face 2] */
adj(A,"pl",0,B,"pl"), adj(B,"pl",0,C,"pl"),
A<>C, A<>B, B<>C.
```

(Note that the first clause in the above rule captures the feature information that A, which is a plane face, has concave (0) adjacency with B, which is also a plane face.)

The above rule is capable of recognizing the existence of a feature belonging to the family of three-sided slots in any object. However, we also need to identify the specific member within the three-sided slot family on the basis of root face to boundary face adjacencies. Again, this information is explicitly available within the feature-MAAG. Thus the following PROLOG rule is easily written:

```
feat(slot,[E,F,G],[A,B],[C,D]):-
/* [[root_feat,[root faces],[entrance faces],[top faces]] */
root_feat(slot,[E,F,G]);
adj(A,"pl",1,E,"pl"),   adj(A,"pl",1,F,"pl"),   adj(A,
                                               "pl",1,G,"pl"),
adj(B,"pl",1,E,"pl"),   adj(B,"pl",1,F,"pl"),   adj(B,
                                               "pl",1,G,"pl"),
A<>B,
adj(C,"pl",1,E,"pl"), adj(D,"pl",1,G,"pl"),
C<>A, D<>A, C<>B, B<>D, C<>D, C<>E,
C<>G, D<>E, D<>G.
```

Note that each of the adjacency clauses in the rule has a one-to-one correspondence with one of the arcs in the feature-MAAG. Hence it is a straightforward and unambiguous task to write the PROLOG rule for a feature by first sketching its feature_MAAG and from this deriving the feature rule. Thus the new approach has provided an effective tool for reducing the expertise required and the ambiguity involved in the process of writing feature recognition rules which was missing in the approach proposed by Henderson and Anderson (Henderson, 1984; Henderson and Anderson, 1984). Following this approach, it is a simple task now to write the rules for recognizing the cylindrical-projection and cylindrical-projection-in-corner-pocket features present in the object illustrated in Fig. 2a. (Hence these rules are not reproduced here.)

Notwithstanding the improvements achieved by the new rule-based approach described above, one general problem concerning ruled-based systems for feature recognition has remained unresolved. This pertains to the fact that it is necessary to write a separate recognition rule for each feature of interest. Therefore it is necessary to create a rule base which anticipates every feature of interest. This could lead to an enormously large rule base. In fact, in principle, the possible number of geometric features is infinite since features can interact to produce new features. To illustrate this point, consider the slot feature in Fig. 2a again. This feature could be called a *primitive feature* since, from a heuristic or functional point of view, it does not make sense to model it as a result of interaction between other simpler features. By a similar argument, the quarter-cylindrical-projection feature in Fig. 2a can be classified as a primitive feature. However, the same is not true in the case of the cylindrical-projection-in-corner pocket feature which would be extracted as a single feature by the feature extraction procedure of Venuvinod and Yuen (1994). Clearly, one would interpret this feature to be a result of a certain kind of interaction between two primitives called the quarter-cylindrical-projection and the corner-pocket. Such features resulting from interactions between primitive features may be called *interacting features*. Thus, although primitive features might be finite in number, one could generate new features ad infinitum by interacting different groups of primitive features in different ways. Further research is therefore required to resolve this problem of feature explosion. Note however that the expert system approach proposed here is capable of recognizing both primitive and interacting features.

The authors have developed and tested software based on the concepts developed in Sections 2 to 4. The input of the system is the DXF file of the object to be analysed. An EWEDS file is then automatically generated from the DXF file. Next, the EWEDS is compressed by using the *same face* procedure (Wong, 1992). The EWEDS is then traversed to find a match (if it exists) with each of the rules in the rule base. (The task of interfacing Venuvinod and Yuen's feature extraction algorithm is still in progress.) The current rule base is organized into five feature libraries. The first two of these store the rules for recognizing root-features: one for polyhedral primitives and the other for cylindrical primitives. The third and fourth libraries are devoted to polyhedral and cylindrical final features (i.e. features including the root-feature, the boundary faces and the root to boundary face adjacencies) respectively. The final library holds the rules for recognizing interacting features. It is intended to progressively expand the rule base by adding rules for new features.

Figure 4 illustrates one of the test objects used which contains 109 faces (as generated while modelling the object in AutoCAD). Note that the object has 14 primitive features and 3 interacting features. All these features have been successfully extracted and recognized, as is evident from the following extract from the output of the software in this test:

sameface(−1,"pl",[13,12,11,9,10])
sameface(−2,"pl",[35,33,32,31,30,29,28,27,26,34,25,
23,24])
. . . .
sameface(−11,"pl",[99,100,98,101,97,96,94,95])
feature(1,"step",[[37,36],[−3,−4],[43,44]])
feature(2,"step",[[39,38,],[−3,−4],[44,45]])
feature(3,"step",[[58,57],[−7,−5],[59,56]])
feature(4,"step",[[60,59],[−7,−5],[−6,58]])
feature(5,"step",[[79,78],[−9,−8],[80,77]])
feature(6,"step",[[81,80],[−9−8],[82,79]])
feature(7,"bl_step",[[103,102,104],[−6,−7,−10],
[−6,−7,−10]])
feature(8,"slot",[[16,15,14],[−2,−11],[21,20]])
feature(9,"slot",[[19,18,17],[−2,−11],[22,21]])
feature(10,"bl_slot",[[4,3,2,1],[−1,−2],[−1,−2]])
feature(11,"bl_slot",[[8,7,6,5],[−1,−11],[]])

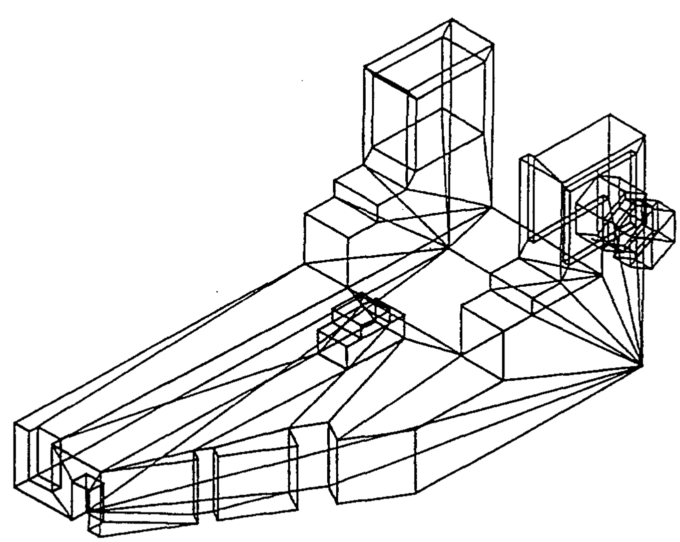

Fig. 4. The AutoCAD model of an object used for testing the new approach.

feature(12,"pocket",[[109,105,106,107,108],[−5],[−5]])
feature(13,"hole",[[74,73,76,75],[−8,−9],[−8,−9]])
feature(14,"slot_cyl",[[111,112,113],[110,114],[110,114]])
Complex_feature(1,"male","tongue",[[1,2],[−3,−4],[44]])
Complex_feature(2,"compound","connected_step",[[3,4],[−7,−5],[58,59]])
Complex_feature(3,"compound","connected_step",[[5,6],[−9,−8],[79,80]])

(Faces with a negative label are the faces obtained by applying the *same face* rule.)

5. Conclusion

The feature recognition procedure described above is adequate for both primitive and interacting features if they can be fully and unambiguously defined by information at the topological and coarse geometry levels. The PROLOG facts based on EWEDS are fully capable of supporting this task. However, additional steps may be needed when more finely defined features are to be recognized. For example, mere topological and coarse geometry information is not adequate to distinguish between the various features illustrated in Fig. 3 since, here, one also requires the magnitudes of the angles between some faces, i.e. fine geometry information.

The new approach, although based on the concept of AAG originally proposed by Joshi and Chang (1988), has the following advantages over their approach:

(1) The ability to recognize a wider range of features—the new methodology enables the recognition of cyndrical features, features with other analytical forms of curved surfaces and features characterized by surfaces adjacencies other than mere concavity or convexity. This has been made possible by the development of the multi-attributed adjacency graph (MAAG) which is an enhanced version of the attributed adjacency graph (AAG) of Joshi and Chang;

(2) The ability to add a new feature to be recognized without disturbing the current structure of the feature recognition process—this has been made possible by the replacement of Joshi and Chang's data-driven procedure by an expert system approach driven by PROLOG rules working in the context of a given set of PROLOG facts which completely define the topology, coarse geometry, and the more important aspects of fine geometry;

(3) The ability to identify the boundary faces of each feature—this has been made possible by including root to boundary face adjacencies in the feature-MAAG.

The new approach has improved upon Henderson's expert system approach (Henderson, 1984; Henderson and Anderson, 1984) by facilitating a more disciplined approach towards the specification of feature recognition rules through the development of feature-MAAGs which are capable of fully capturing the necessary and sufficient conditions for feature recognition. Further, the adoption of Venuvinod and Yuen's feature extraction algorithm (Venuvinod and Yuen, 1994) enables feature extraction to be carried out prior to recognition. This reduces the computational load in traversing the rule base.

One problem concerning the new methodology however needs to be noted. This pertains to the need for developing a large rule base and the resulting computational explosion during the feature recognition process. The resolution of this problem needs further research.

Acknowledgements

The authors thank the Department of Manufacturing Engineering, City University of Hong Kong for the facilities provided to conduct this study.

References

Baumgart, B.G. (1974) Winged-edge polyhedron representation: Technical Report Stan-Cs-320, Computer Science Dept, Stanford University, Stanford, CA.

Biederman, I. (1985) Human image understanding: recent research and a theory, in *Human and Machine Vision II*, Academic Press, London, pp. 13–57.

Boothroyd, G. and Dewhurst, P. (1983) Design for assembly, Department of Mechanical Engineering, University of Massachusetts.

Henderson, M.R. (1984) Extraction of feature information from three-dimensional CAD data, Ph.D. thesis, Purdue University.

Henderson, M.R. and Anderson, D.C. (1984) Computer recognition and extraction of form features: a CAD/CAM link. *Computers in Industry*, **5**, 329–339.

Joshi, S. and Chang, T.C. (1988) Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-Aided Design*, **20**(2), 58–86.

Mattei, D. (1993) New version of IGES supports B-Rep solids. *Mechanical Engineering*, January, 50–52.

Venuvinod, P.K. (1993) Automated analysis of 3D polyhedral assemblies: assembly directions and sequences. *Transactions of NAMRI/SME*, **XXI**, 399–405. Also published under the same title in (1993) *Journal of Manufacturing Systems*, **12**(3), 246–252.

Venuvinod, P.K. and Yuen, C.F. (1994) Efficient automated geometric feature recognition through feature coding, accepted for publication. *Annals of the CIRP*, **43** (3), 413–416.

Wong, S.Y. (1992) Automated geometric feature recognition—an essential component for automated design for assembly, M. Phil. Thesis, City Polytechnic of Hong Kong.