Efficient Automated Geometric Feature Recognition through Feature Coding

J 28

P. K. Venuvinod, C. F. Yuen, City Polytechnic of Hong Kong – Submitted by M. E. Merchant (1)

Received on January 6, 1994

Summary: Automated geometric feature recognition (GFR) is a commonly encountered task in the creation of any process planning or design for manufacturing software. This paper describes a new method based on feature coding for automated GFR. An enhanced winged edge data structure including surface type labels and a Multi-Attributed Adjacency Matrix (MAAM) is generated from the CAD model of the given object. The MAAM fully captures the topology and coarse geometry of the object for the purposes of GFR. A simple algorithmic method extracts each feature from the object-MAAM. The feature-MAAM is then processed to generate a unique code which is recognised and interpreted by matching it with entries in a Feature Database. The method is significantly superior to previous GFR methods in terms of computational efficiency and the reduced need to invoke expert rules. Unlike previous methods, the system can handle objects with plane, cylindrical as well as other analytically definable curved faces and can recognise both simple and complex features i.e. those formed by interactions amongst simple features.

Keywords: Feature recognition, Feature Coding, Process Planning

1. INTRODUCTION

Concurrent engineering is emerging as the contemporary solution to the ever increasing demand in the world market for reducing the lead times involved in the design and manufacture of products. Thus, much international research effort has recently been invested into the development of a range of CAPP (Computer Aided Process Planning) and DFX (Design for manufacture, assembly, etc., i.e. X) software which enable quick evaluation of each design conception and improvement in terms of ease of manufacture and the resulting manufacturing cost.

Today, most CAPP and DFX activities are being conducted manually with or without the aid of computerised procedures and databases. For example, the well known UMass system [3] for Design for Assembly (DFA) has now a rudimentary computerised version. However, the application of such software is quite tedious which discourages their extensive use so that further computerisation seems to be needed to accellerate the adoption of concurrent engineering. However, since the full computerisation of all the professional tasks involved in these activities is a mammoth exercise, one has to first address the question of rationally prioritising the tasks in the order of the desirability of their computerisation.

Carr et al have recently examined this issue from a humanistic point of view in the general context of knowledge based systems [4]. In particular, they observed that the tasks undertaken by any professional can be divided into intrinsic and extrinsic types. Intrinsic tasks are those that give professional pride and job satisfaction. The professional could feel threatened if these jobs are taken away by computerisation. Extrinsic tasks are those that are tedious and unsatisfying. The professional would be happy to relegate these to a human assistant or a computer. Thus the automation of any professional activity should focus on the extrinsic tasks and leave the intrinsic tasks alone.

Consider now the task of feature recognition which is commonly encountered in CAPP and DFX activities. In particular two types of feature recognition are encountered: geometric and technological. For instance, in CAPP for machining, the recognition of a "pocket" in the CAD model of the part to be machined is likely to lead to the decision of invoking a pocket milling canned cycle in the NC program. This involves geometric reasoning. Likewise, in the application of the UMass system of DFA, it is often necessary to recognise if the parts are likely to stick together due to magnetic force or grease coating. This involves technological reasoning.

Now, it can be argued that Geometric Feature Recognition (GFR) is a tedious task and, hence, may be classified as extrinsic. Further, there appears to be much in common in the geometric reasoning involved in diverse CAPP or DFX tasks. In contrast, technological feature recognition appears to be intrinsic and domain specific (i.e. it is specific to the particular product family, process or, even, industry). Hence it is suggested that GFR should be accorded high priority in the automation of CAPP and DFX activities.

Several strategies for GFR have been reported in the literature. These include approaches based on syntactic pattern recognition [11,13], automata [8], decomposition [6], graph theory [5,9], the convex decomposition technique [10], rule based expert systems [7,14], or combinations of these. (Since the literature on the subject is extensive, only a few important publications are referred to here.) However, each of these approaches is found to be associated with one or more of the following disadvantages:

- limitations on the types of features that can be recognised (most works are limited to polyhedral and/or cylindrical features),
- the need for extensive computations,
- difficulties in expanding the features database to include new features as they are encountered, and
- the fact that the specification of the expert rules, if any, is itself an expert task.

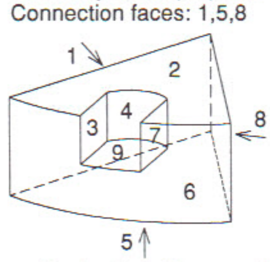
The present paper reports on a new approach based on feature coding with a view to overcoming the above limitations.

2. SOME BASIC CONCEPTS CONCERNING GFR

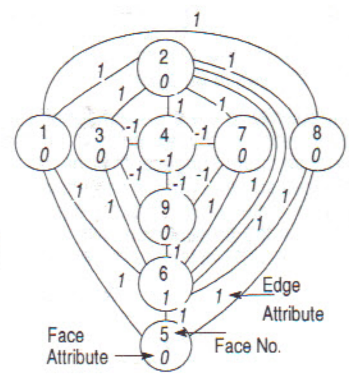
A general feature is defined as a functional entity (object, shape, or process) which is meaningful in certain context [12]. In the context of GFR, a geometric feature may be defined as a descriptor of the subset of the geometric model of a part whose presence is relevant in a given context. A generic GFR system should be capable of (i) facilitating the recognition of the wide variety of geometric features encountered in diverse contexts, and (ii) adding new features, as they are encountered, to the feature database for the particular context.

A review of literature suggests that the information required for GFR can be classified into three levels: Topology, Coarse Geometry, and Fine Geometry.

Root faces (RF): 3,4,7,9
Boundary faces (BF): 2,6
Connection faces: 1,5,8



(a) An object with an edge pocket



Notes:
Face attributes:0 for plane type; 1 for convex and -1 for concave cylindrical surface respectively.
Edge attributes:0 for smooth; 1 for convex and -1 for concave edge respectively.

(b) Object MAAG

Face	1	2	3	4	5	6	7	8	9
1	0	1	9	9	1	1	9	1	9
2	1	0	1	1	9	1,1	1	1	9
3	9	1	0	-1	9	1	9	9	-1
4	9	1	-1	-1	9	9	-1	9	-1
5	1	9	9	9	0	1	9	1	9
6	1	1,1	1	9	1	1	1	1	1
7	9	1	9	-1	9	1	0	9	-1
8	1	1	9	9	1	1	9	0	9
9	9	9	-1	-1	9	1	-1	9	0

(c) Object-MAAM

	BF	RF	RF	BF	RF	RF
Face	2	3	4	6	7	9
2	0	1	1	9	1	9
3	1	0	-1	1	9	-1
4	1	-1	-1	9	-1	-1
6	9	1	9	1	1	1
7	1	9	-1	1	0	-1
9	9	-1	-1	1	-1	0

9 represents non-adjacency

(d) Feature-MAAM

Figure 1: Example of a prismatic object with an edge-pocket feature

Topology is concerned with the adjacency relationships existing amongst the faces (F), edges (E), and vertices (V) composing the object. This information is often succinctly captured by using Adjacency Graphs (AG) or the corresponding Adjacency Matrix (AM). A widely used graph in GFR is the Face-Edge (F-E) Graph in which each node represents a labelled face and each arc connecting a pair of faces represents the fact that the two faces have a common edge in the object.

Coarse geometry is concerned with the geometrical attributes of the nodes and arcs of the adjacency graph. In the case of F-E graphs, this pertains to the attributes of the faces and edges. For instance, Joshi and Chang have used an Attributed Adjacency Graph (AAG) where each arc (edge) is associated with an attribute-label denoting whether it is concave or convex [9]. However, this approach is applicable only to polyhedral objects or to polyhedral approximations of objects with curved faces. In order to overcome these limitations, the present paper proposes the use of the Multi-Attributed Adjacency Graph/Matrix (MAAG/MAAM) where the edges as well as the faces carry attributes.

Figure 1a shows a simple prismatic object with an edge-pocket feature. Figures 1b and 1c show the corresponding MAAG and MAAM respectively which are obtained by applying the attribute-labelling scheme. The following general properties of a MAAM may be noted :

- it is a symmetric matrix capturing both the topological and coarse geometry properties of the object,
- the diagonal cells contain the face attributes,
- the off-diagonal cells contain the edge attributes, and
- while there should be a single entry in each diagonal cell, there could be more than one entry in the off-diagonal cells (as in the case of faces 2 and 6 in Figure 1a which have two common edges).

Fine Geometry refers to information concerning the precise dimensions, angular orientations, tolerances, etc. of the faces, and the internal relationships, such as parallelism, perpendicularity, etc. within specified face sets. However, in practice, feature recognition involving information at the fine geometry level (unlike that at higher levels) tends to be highly context specific. Hence, the scope of the present paper is limited to GFR at the topological and coarse geometry levels.

Even at the levels of topology and coarse geometry, the conceivable number of features is infinite because features with a certain degree of complexity can interact to produce more complex features, and this can go on *ad infinitum*. For instance, two intersecting slots cutting across a face produce a complex feature with four protrusions. Hence, in principle, it is unrealistic to aim at recognising every possible feature. However, within any given context, the number of features of interest are a subset (not necessarily finite) of this infinite universal feature set. Further, it is unlikely that the context specific user would know, *a priori*, all the features to be recognised. Hence, the structure of a generic GFR system should be so organised as to enable the incremental addition of each new feature, as it is encountered in the particular context, to the feature database.

A prerequisite to the development of a rational structure of GFR is an understanding of the taxonomy of the features to be recognised. A review of literature suggests that this issue has not been systematically addressed so far. Even the nomenclature used in labelling the features has been quite varied. Figure 2 illustrates the taxonomy of features as adopted in the present work with a view to facilitating the incremental expandability of the feature coding database.

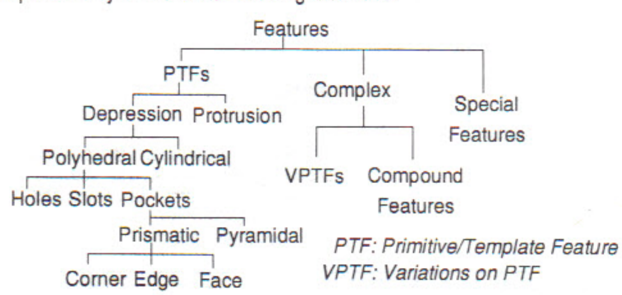


Figure 2: Taxonomy of features

There are three basic steps involved in GFR: feature extraction, feature identification, and feature interpretation.

Feature extraction is concerned with identifying the object faces, edges and vertices associated with each individual feature. Note that some of the faces may be common to more than one feature and some may not belong to any feature (the latter may be called connection faces). As pointed out by Joshi and Chang [9], when the F-E graph is used, feature extraction reduces to a process of identifying the sub-graph associated with each feature. In particular, they identified the subgraphs on the basis of the heuristic that "a face that is adjacent to all its neighbouring faces with a convex angle does not form part of a feature". In other words, the root faces of a feature may be extracted by decomposing the object at concave edges (a root face is defined as a feature face with one or more concave edges). It is interesting to note that human beings seem to adopt a similar process while recognising objects. Biederman has recently reported empirical evidence which suggests that the first step in human image understanding is the decomposition of the object at points of deep concavity [2]. Arguably, since the cognition processes used by humans have evolved over thousands of generations through the principle of "survival of the fittest", these processes must be inherently robust. Hence, the feature extraction strategy adopted in the present work is based on the principle of concave decomposition. However, the new feature extraction strategy, unlike that adopted by Joshi and Chang [9], is capable of extracting both the root and boundary faces of each feature (see section 4).

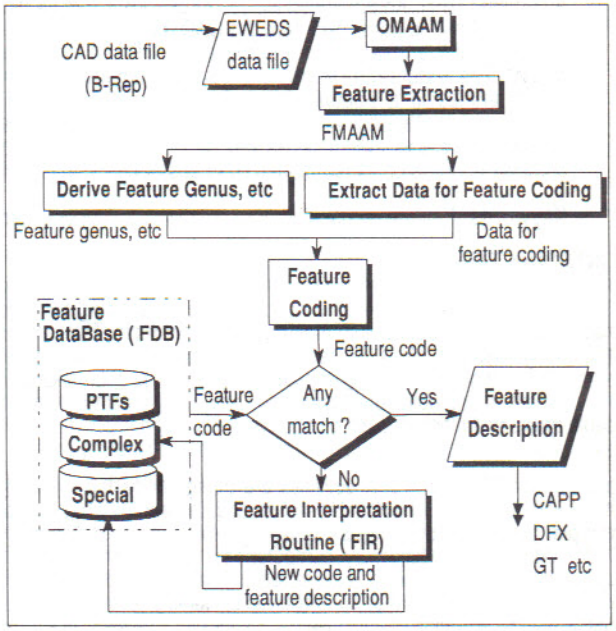


Figure 3: Flowchart of GFR through coding system

A wide variety of feature identification strategies has been reported in the literature. Of these, the expert system approach and the graph based approach are of particular interest. Typically, in the former approach, the topological and coarse geometry properties of each likely feature are coded as a set of rules and an exhaustive search is conducted in the object model for the presence of a feature satisfying the rule set. This approach, in principle, satisfies the condition of incrementally expandable GFR. However, the method is computationally expensive since an exhaustive search has to be conducted with respect to each feature available in the feature database. Joshi and Chang [9] avoid this exhaustive search by first decomposing the object AAG into subgraphs corresponding to each feature and, then, applying a highly structured procedure, called the Recogniser, to identify the feature. However, this advantage is realised at the cost of expandability since the structure of the Recogniser procedure needs to be modified if a new feature is to be accommodated. A different approach based on feature coding is proposed in the present work with a view to achieving expandability at a low computational cost. Figure 3 provides an overview of this new approach. The following sections explain the major steps involved.

3. EWEDS AND OBJECT-MAAM

It is assumed that the object has been modelled using a CAD system based on Boundary Representation (B-Rep) and that each pair of isoparametric adjacent faces with a smooth common edge has been merged into a single face. In the current implementation, AUTOCAD has been used. A specially written program automatically converts the information contained in the Data Interchange File (DXF) output by AUTOCAD into the Enhanced Winged Edge Data Structure (EWEDS) [14]. EWEDS is an enhancement of the well known Baumgart's Winged Edge Data Structure (WEDS) [1]. WEDS, in its original form, was confined to polyhedral objects. In order to enable the processing of non-polyhedral objects too, EWEDS includes several enhancements. EWEDS is presented in terms of vertex, edge and face types of PROLOG unit clauses:

- *edge(edge_number, vstart, vend, fcw, fccw,* ***angle****, ncw, pcw, nccw, pccw)*
- *vertex(vertex_number,* ***first edge_list****, coordinates)*
- *face(face_number,* ***first_edges, type_of_face, parametric_data_list_list****)*

where the fields in **bold** font are the enhancements, *vstart* and *vend* are the terminal vertices of the edge, *fcw* and *fccw* are the clockwise and counter-clockwise faces, ***angle*** is the angle between *fcw* and *fccw* on the material side, *ncw* and *pcw* are the next and previous clockwise edges, *nccw* and *pccw* are the next and previous counter-clockwise edges, ***first_edges*** point to the number of edge-loops in a face, ***type_of_face*** follows the face attribute labelling scheme, and ***parametric_data*** includes information concerning the orientation of the face.

It is a straight forward exercise to write a simple routine to generate the object-MAAM from the EWEDS file.

4. FEATURE EXTRACTION AND FEATURE-MAAM

The new extraction procedure aims to identify the root and boundary faces associated with each feature in the object. The object-MAAM is scanned row by row starting from the first row. The first "-1" entry in any cell triggers a feature extraction cycle. There are two cases to be considered.

Consider first the case when the extraction cycle is triggered by an off-diagonal cell, (i,j) with $i \neq j$, which indicates the presence of a concave edge. Then faces F_i and F_j are flagged as root faces in the current feature. Next, the row for face F_i is scanned for further off-diagonal "-1" entries and, if any, the faces indicated are added to the list of root faces. In addition, the cells with a "1" entry (indicating a convex edge) are noted and the corresponding faces are flagged as boundary faces (a boundary face is an object face which is not a root face and is convex adjacent to one or more root faces of the feature under consideration). This process is repeated for every row corresponding to a previously recruited root face. The cycle is terminated when no new root face is found.

The second case corresponds to the situation when there is a negative entry in a diagonal cell (indicating the presence of a concave face). For instance, such a situation is obtained when a slot with a single concave cylindrical face cuts across another object face. However, note that, in-practice, such a face would constitute a slot feature only if the concave curved face has only convex adjacency with each of its boundary faces. Thus the presence of a negative entry in a diagonal cell is allowed to trigger a feature extraction cycle only when there are no "0" or "-1" entries in the row corresponding to the cell. Note that, on this basis, face 4 by itself in Figure 1a does <u>not</u> constitute a feature.

It is a straight forward exercise at this stage to construct the feature-MAAM by extracting from the object-MAAM the rows and columns corresponding to the root and boundary faces flagged during the feature extraction cycle but ignoring the adjacency relationships between pairs of boundary faces. Figure 1d shows the feature-MAAM thus obtained for the edge-pocket feature in Figure 1a.

Having extracted one feature, the next feature, if any, is extracted by repeating a similar procedure with the exception that a cell corresponding to a face previously flagged as a root face is not allowed to trigger a feature extraction cycle. This procedure is repeated until no new feature can be extracted which happens when (i) there is no negative entry in any off-diagonal cell and (ii) there is a negative entry in a diagonal cell only when associated with one or more 0 or -1 entries in the corresponding row.

5. FEATURE CODING

All the topological and coarse geometry properties necessary for feature identification are easily derivable from the feature-MAAM and EWEDS. Traditionally, in the expert system approach [7, 14] these properties are captured in the form of a PROLOG rule for the feature, and a search is made in the entire object model for a feature satisfying the rule. Such an expert system approach, although satisfying the condition of incremental expandability of the feature database, has many disadvantages. Firstly, an expert rule needs to be written for each and every feature of interest. Secondly, it is computationally expensive. If there are N_d features in the database and N features in an object, the number of times the object-MAAM needs to be traversed may reach $N \times N_d$. The process of feature extraction as described in section 4, reduces this maximum number of traversals to N.

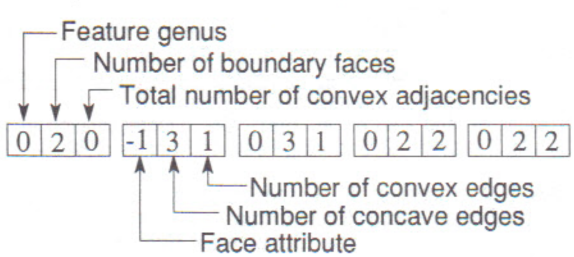


Figure 4: Feature code of example in figure1a and its coding method

The computational load could be further reduced if the essence of the feature-MAAM could be compressed into a code. The codes for all the features of interest in a specific context could then be stored in the feature database as a look-up table so that the feature identification process is reduced to one of merely traversing the look-up table. A prerequisite to this approach however is a feature coding system which (i) is able to capture the essences of feature-MAAMs, (ii) yields a unique code for any given feature, and (iii) is robust, i.e. is unlikely to require substantial modifications when new features are added to the feature database. Figure 4 shows the code generated for the edge-pocket feature in Figure 1a, as well as its coding method which, it is believed, satisfies the above requirements (the authors had explored several alternatives before deciding upon this particular coding method).

The proposed code is a polycode of variable length. The code length is 3(n+1) where n is the number of root faces in the feature being coded. The code digits are organised in groups of three.

The first digit in the first group is the genus, g, of the open object formed by the root faces. Genus is the topological name for the number of passages piercing through the object and is calculated by using the well known Euler's formula for open objects: $g=b-f+e-v+l$, where b is the number of bodies assumed equal to 1 (since one feature is under consideration); and f, e, v, and l are the numbers of faces, edges, vertices, and inner loops respectively in the open object. f is equal to the number of faces flagged as root faces during the feature extraction cycle. e is total number of edges in the open object. e, v and l are easily determined from the EWEDS. For instance, note that l equals the total number of *first_edges* less the number of faces in the open object.

The second and third digits in the first group are equal to the number of boundary faces in the feature and the total number of convex adjacencies existing between pairs of root faces respectively. These are easily determined from the feature-MAAM.

Each subsequent group starting from the second group is associated with a specific root face and records the important properties of the root face: the first digit is the attribute label of the root face whereas the second and third digits are taken equal to the numbers of concave and convex edges respectively bounding the root face. This information is easily obtained from the feature-MAAM. The groups following the first group are arranged in the decreasing order of the number formed by the second and third digits of each group. This ordering method significantly reduces the possibility of generating an identical code for what are, essentially, distinct features.

6. PRIMITIVE/TEMPLATE FEATURES (PTF)

It has been noted that new features can be formed by interacting two other features. For instance, Figure 5 shows an object with a slot which interacts with a corner-pocket. The feature extraction procedure given in section 5 will extract this as a single compound feature. However, intuitively, it is seen that this compound feature can be decomposed into two primitive features: a slot and a corner pocket. A primitive feature can be defined, from a functional point of view, as the feature which enable the interpretation of other features at the coarse geometry level.

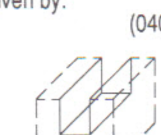
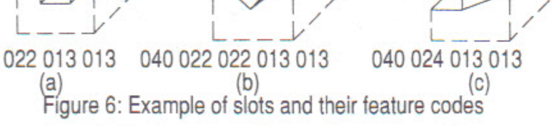


Figure 5: An object with a compound feature

However, there are infinite number of primitive features. For instance, one can conceive of an infinite number variations of the slot feature by varying the number of root faces. See Figures 6a and 6b. Now, it can be shown that the feature code for a polyhedral slot with n root faces and the minimum possible number of convex edges on each root face is given by:

(040) (022)...n-2 times (013) (013)



040 022 013 013   040 022 022 013 013   040 024 013 013
(a)   (b)   (c)

Figure 6: Example of slots and their feature codes

We will call a primitive feature with the minimum possible number of convex edges on each root face as a primitive/template feature (PTF). Note that the feature codes for the slots in Figures 6a and 6b fully conform with the coding formula given above. Hence these two slot features belong to the primitive/template class. Consider now the feature code for the corner-to-corner-slot in Figure 6c which seems to be a variation of the feature code for the slot with three root faces shown in Figure 6a. In particular, it is seen that the variations are confined to the digits denoting the number of convex edges in each root face. We will call such features as Variations on PTFs (VPTFs).

7. FEATURE DATABASE (FDB)

It has already been noted that a feature database makes sense only in a context. Further, the database should be incrementally expandable as new features are encountered in the particular context. From this point of view, it is useful to divide the database into three parts as follows:

1. PTFs: this part will contain a finite list of PTFs along with the coding formulae.
2. Complex features: this will contain features which can be either identified as a variation on a specific PTF, or decomposed into two or more PTFs (or VPTFs) contained in part 1.
3. Special features: This part contains the features that do not belong to parts 1 and 2.

It is desirable that every entry in the database stores the feature code and one or more semantic interpretations which are relevant to the context of the database.

Given a feature database, the feature identification process reduces to one of merely matching the code of the feature in question with each of the entries in the feature database. This simple table-look-up-process makes the new methodology of GFR substantially more attractive than previous GFR methods in terms of the computational load.

The following procedure may be used in incrementally expanding the feature database:

1. Determine the feature code of the new feature.
2. Check if the code satisfies the coding formula for any one of the PTFs in part 1. If it does, there is no need to add the new feature to the database. If it does not, examine if it can be interpreted as a VPTF or decomposed into two or more PTFs (or VPTFs). Note that sometimes it is possible that a feature can have multiple interpretations. The possible and desirable interpretations should be intuitively apparent to the user since he has the full object model available and he knows the context. If an interpretation in terms of PTFs is found feasible, the new feature code along with the semantic interpretation is added to the complex features list. If not, it is added to the special features list.

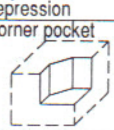
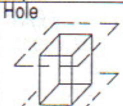
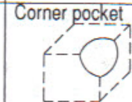
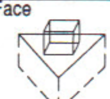
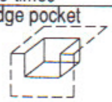
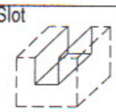
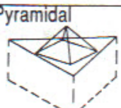
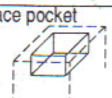
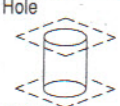
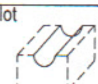
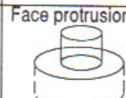
Polyhedral protrusion	Polyhedral depression	Polyhedral depression	Cylindrical
Corner  04(n-2) 0(n-1)3 (013)..n-1 times	Corner pocket  030 0(n-1)2 (031)..022 022 n-3 times	Hole  120 (022).. n times	Corner pocket  030 -103
Face  14(n-1) 0(n-1)3 (013)..n-1 times	Edge pocket  for n=3 020 022 022 021 for n > 3 020 0(n-1)1 (031).. 022 022 n-3 times	Slot  040 (022).. 013 013 n-2 times	2-face pocket  010 -111 010
Pyramidal  03(n-1) 0(n-1)0 (012)..n-1 times	Face pocket  for n=4 010 031 031 031 030 for n > 4 010 0(n-1)0 (031).. n-1 times	Cylindrical Hole  120 -102	3-face pocket  010 -131 031 020
	Pyramidal  010 (021).. n times	Slot  040 -104	Face protrusion  120 111 011

Table 1 : Preliminary list of PTFs

Clearly, it is desirable to minimise the number of features that need to be listed in part 3. This can be done by maximising the number of PTFs on the basis of a rationalised taxonomy of features. Table 1 details the preliminary list of PTFs arrived at by the authors based on such considerations.

8. FEATURE INTERPRETATION ROUTINES

It is believed that a predominant proportion of the features deemed to be complex could be grouped such that the features in each particular group satisfy a specific complex coding formula. To illustrate this point, imagine a face with six identical cylindrical protrusions. The feature extraction rule given in section 4 will extract this configuration as a single feature. However, during the feature identification process, it would be identified as a complex feature. The user can then write a Feature Interpretation Routine (FIR), which could be algorithmic or rule based, to develop a general formula for complex features decomposable into n' number of cylindrical protrusions on a common face. The resulting FIR can then be added to the complex features section of the database. Such an approach will lead to a significant reduction in the size of the database.

9. CONCLUSION

A new methodology for GFR has been developed in the previous sections. The methodology covers the processes involved in the extraction, identification as well as interpretation of features. The inclusion of face attributes in the procedure has enabled the recognition of a wider variety of features than that reported in previous works. Unlike previously suggested methods for feature extraction [8], the method is capable of extracting the root faces as well as the boundary faces. Further, in contrast to previous GFR strategies based on the expert systems approach, the new strategy does not require an exhaustive search through the entire object model each time a new feature is to be identified. Instead, the object-MAAM is broken right at the beginning into feature-MAAMs. The feature codes are then derived from the feature-MAAMs, and the codes are matched against coding formulae available in the feature database. The computational load involved in GFR is thus substantially reduced. The new methodology also facilitates the incremental expansion of the feature database and the provision of semantic interpretation of features in a manner consistent with the context.

Future refinements of the methodology may be directed towards the development of (i) a rational and generalised feature taxonomy with the aim of generating an exhaustive list of primitive/template features, (ii) an array of generalised complex feature interpretation routines, and (iii) a structured syntactic approach towards feature interpretation.

REFERENCES

[1] Baumgart, B. G.,1975, "A Polyhedron Representation for Computer Vision," AFIPS Proc. National Computer Conference.

[2] Biederman, I.,1985, "Human Image Understanding : Recent Research and a Theory," Human and Machine Vision II, Academic Press Inc., 13-57.

[3] Boothroyd, G., 1987, "Design for Assembly and Design for Manufacture," Annals CIRP, **36**, 2, 435-437.

[4] Carr, C., Jeffrey, H. J., and Putman, A. O.,1992, "Knowledge Based Systems and Intrinsic Practices," Expert Systems with Applications, **4**, 4, 239-247.

[5] de Floriani, L.,1987, "A Graph Based Approach to Object Feature Recognition," ACM, **6**, 640-658.

[6] Grayer, A. R.,1977, "The Automatic Production of Machined Components Starting from a Stored Geometry Description," Advances in Computer-Aided Manufacture, Ed. McPherson, D., North Holland Pub. Co., 137-152.

[7] Henderson, M. R.,1984, Extraction of Feature Information from Three-dimensional CAD Data, PhD Thesis, Purdue University.

[8] Iwata, K., Kakino, Y., Oba, F., and Sugimura, N.,1980, "Development of Non Part Family Type Computer Aided Process Production Planning System," Advances in Manufacturing Technology, Ed. Blake, P., North Holland Pub. Co., 171-184.

[9] Joshi, S.,and Chang, T. C.,1988 "Graph Based Heuristics for Recognition of Machined Features from a 3D Solid Model," Computer-aided Design, **20**, 58-66.

[10] Kim, Y.S.,1992, "Recognition of Form Features Using Convex Decomposition," Computer-aided Design, **24**, 9, 461-476.

[11] Kypriyanou, L. K.,1980, Shape Classification in Computer Aided Design, PhD Thesis, University of Cambridge, U. K.

[12] Shipitlani, M. and Fischer, A., 1991, "CSG Representation as a Basis for Extraction of Machining Features," Annals CIRP, 40, 1, 157-160.

[13] Staley, S.M., Henderson, M. R., and Anderson, D. C.,1983, "Using Syntactic Pattern Recognition to Extract Feature Information From a Solid Geometric Data Base," Computers in Mech. Engg, 61-66.

[14] Wong, S. Y., 1992, Automated Geometric Feature Recognition, MPhil. Thesis, City Polytechnic of Hong Kong.