Transactions on Engineering Sciences vol 16, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

Application of feature recognition from solid models to CMM technology

R. Tuttle*, G. Little*, D.E.R. Clark** and J. Corney*

**Department of Mechanical & Chemical Engineering,*
***Department of Mathematics, Heriot-Watt University, Riccarton Campus, Edinburgh - SCOTLAND, EH14 4AS*

Email: b.tuttle@hw.ac.uk

Abstract

Recent research has produced a suite of powerful generic algorithms for identifying features on solid models, and after further refinement they are now capable of analysing the geometry of complex engineering components. In addition, a graphical user interface (GUI) has been developed for human-computer interaction. The main algorithm has also been embedded in a commercially available computer aided manufacturing (CAM) package which assists the manufacturing engineer in the production of NC code for machine-tools.

A second application for the feature recognition engine is now being investigated, namely that of interaction with co-ordinate measuring machine (CMM) technology. Recent research has indicated that comparing a manufactured component (or batch of components) with a computer generated solid model gives a quick and highly accurate set of deviation data. Similarly, a solid model can be constructed from CMM probe data which represents the component tested. Identification of significant features within the solid model further assists the process by allowing the CMM operator to select areas of importance which have been suggested by the feature recognition system.

1. Introduction

CMM technology has advanced rapidly over the past decade. When coupled with modern day micro-computers for computer numerical control (CNC) the

CMM is now capable of far more than simply *off-line* post process inspection. In-process gauging is increasingly an objective within the concept of manufacturing *right first time*. Much CMM software now allows the end user to operate in a *mixed model mode*, achieving manipulation and analysis of component topology through solids, wireframe and surface modelling as appropriate, in a hybrid *modelling environment*.

Additionally, the use of feature recognition on the solid models within such an environment will give extra data which can aid the end-user and, ultimately, reduce both development and manufacturing times through process simplification, streamlining and the elimination of the artificial walls between design, manufacture and metrology.

Automated recognition of geometric features, based upon Boundary Representation (B-Rep) solid models, has been developing for over two decades, evolving from seminal work by Kyprianou [1] on shape classification in Computer Aided Design (CAD). He showed that a loop of *convex* edges on a solid model indicated the presence of a *depression* feature, whilst a loop of *concave* edges signalled a *protrusion* feature.

Since then a number of researchers have developed a range of methods of feature recognition each tailored to a particular application. Two of the more productive areas of research are the *volumetric* and the *graph-theoretic* approaches. The former approach has been developed typically in the work of Kim et al [2] and Sakurai [3],[4] whilst the latter method has been brought to fruition in the work of such researchers as DeFloriani [5] and Sormaz et al [6]. Almost without exception, these feature recognition systems identify the features as *face sets*. The methods for achieving this vary dramatically, but the outcome is essentially the same - namely a feature on the component either wholly or partly bounded by a closed cycle of faces and/or edges.

Corney and Clark [7] used a graph-based methodology to analyse solid models with respect to a specific machining direction. Their work was restricted by the assumption that the component be single-sided, i.e. the component may be completely machined from one tool approach direction. This work has since been extended by Tuttle et al [8] encompassing increasingly complex geometry for applications in both computer aided manufacturing (CAM) and numerical control (NC) machining.

It would appear that, until recently, very little of this plethora of feature recognition research has left the confines of research establishments and literature. Recently however, several commercial products have appeared which are either self-contained applications, based on feature recognition from pre-defined libraries (such as the PART system [9] for process planning), or are capable of identifying user-defined patterns (templates) of geometry and topology in isolation from any particular application [10]. These types of technology also offer methods of linking information about a component's shape with downstream activities such as cutter path generation or process planning, manufacturing schedules and co-ordinate measuring machines (CMMs).

2. CMM Technology and Features

Several noteworthy areas of research investigate the relationships between component topology, form features and CMMs. Roy [11] studied the geometric characteristics of manufactured components in the development of algorithms for the inspection software of CMMs. He concluded that there was a necessity to develop rigorous definitions of features in order to avoid any ambiguities. Pahk et al [12] also linked *simple* computer aided design (CAD) defined features such as holes, slots and bosses for CMM applications and developed a prototype measurement system for manufacturing moulds based on the work. A promising neural network approach to determining optimal sample size of *hole* features was taken by Zhang et al [13] and an implicit correlation between sample size and feature complexity was found.

Research in directly linking CAD data to CMM technology includes the investigations of Merat and Radack [14] who concentrated on the automated generation of inspection plans for CMMs by requiring that the original CAD model to be built in terms of *form features*. A *comparative analysis* module for tolerancing information used to support subsequent process planning, manufacturing scheduling and CMM inspection was investigated by Ge et al [15]. The proposed module allowed tolerancing information to be directly specified for a 3D CAD model of a component and furthermore allowed this information to be attached to a *single feature* as well as to a group of *same pattern features.*

It is important to note that, without exception, the work mentioned above relies upon the existence of pre-defined libraries of features, in some cases incorporating rigorous rule bases, in order for feature recognition to be applied to CMM technologies. This necessity for a feature library is a severely restricting factor. For example, the problem concerning the nature or definition of a feature is well illustrated in figure 1. Perusal of this figure prompts a number of important questions. Under what conditions can two protrusions be considered as a single slot? When should a slot be considered as an open pocket? How does one quantify the 'interactivity' of slots and protrusions? Which rule-base is correct?

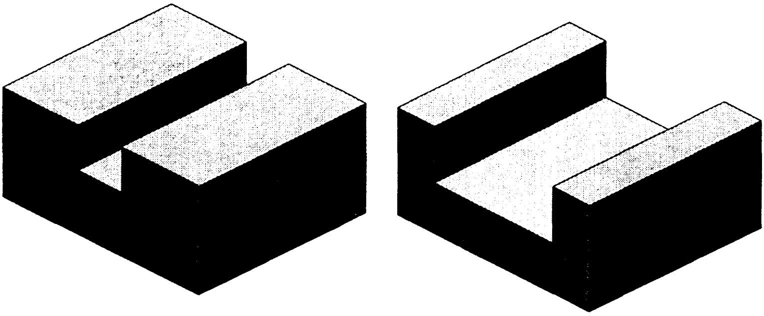

Figure 1. Slot/protrusion problem

Feature *interaction* is illustrated in figure 2. This configuration may be considered as two crossing and interacting slot features, four touching slot features, one crossing slot feature with two touching slot features, a cross-shaped depression feature or as four separate protrusion features!

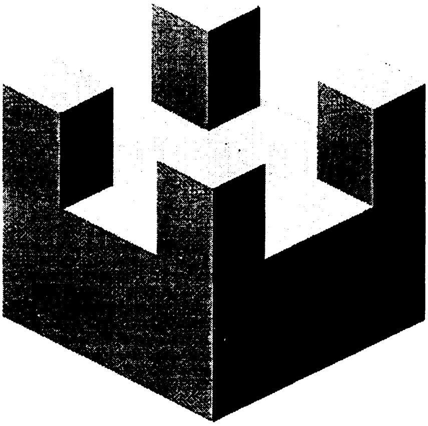

Figure 2. Increased slot/protrusion complexity

Thus, any feature library is limited by the ability of its contents to represent protrusion or depression features on the component in question and the way in which we choose to answer the above questions. Even for examples as simple as those shown, ambiguity occurs, and for more complex components and associated geometry the restriction can quickly become insurmountable.

3. Feature Recognition without a Feature Library

3.1 Path Generation

The graph-theoretical approach of Corney and Clark [7] considered feature recognition from the specific viewpoint of NC machining. Here the features of a model became associated with the machining direction, or *aspect direction,* **d**. Their theory was applicable to models containing only planar faces and furthermore omitted all non-vertical edges and faces from the face-edge graph leaving, what was termed, an *aspect face-edge graph.* This graph was then searched for closed cycles whilst checking that the orientation of the traversals of the faces was consistent. The condition was imposed that any traversal of a face should be in the direction **n** x **d** (where **n** is the outward normal of the face). This was implemented by creating a vector, **v**, from the start position of the edge connecting the face to the previous face to the start position of the edge connecting the face to the next face in the cycle (Figure 3). If **v**.(**n** x **d**) is positive then this traversal is valid. As a result of this principle, depression cycles are found with a clockwise orientation and protrusion cycles with an anti-clockwise orientation with respect to **d**.

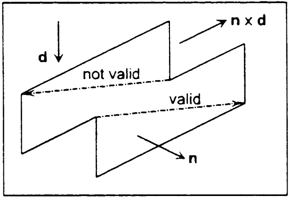

Figure 3. Original path creation

It was noted that using the Corney/Clark algorithm, when stepping off a more complex face, perhaps half of the possible routes considered lead to *invalid* cycles. This is the source of one of the inefficiencies of their algorithm. The search method has since, therefore, been considerably modified. In the ACIS[a] solid modeller, a face is bounded by a *loop* of *coedges* which are directed instances of the edges adjacent to that face. Viewed from the exterior of the model, the boundary of a face consists of a list of coedges which form an anti-clockwise loop and any internal boundary of the face takes the form of a clockwise loop. A path search criteria was adopted whereby the coedges of a face are divided into those which may only be used to step *onto* the face and those which may only be used to step *off* it. These coedges are classified as *on-edges* and *off-edges* respectively. An *on-edge* of a face is defined as one whose coedge is in the direction **d**, and an *off-edge* is one whose coedge is in the direction **-d** (see

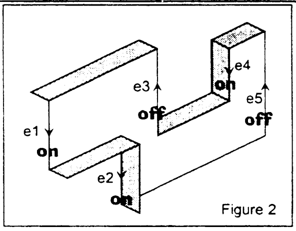

Figure 4. Directed path creation

Transactions on Engineering Sciences vol 16, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

figure 4). A path may only be created from an on-edge to an off-edge. Any other path, such as one from an on-edge to an on-edge, is invalid.

The orientation of a path across a *cylindrical* face is more complex. On a plane face, a path is said to be *correctly oriented* if it is in the direction **n** x **d**. But the normal direction varies around a cylindrical face, so this condition is inapplicable. Instead, here the rule is that if the body is *internal* to the cylinder then a path should travel anti-clockwise when viewed from the approach direction, and, if the body is *external*, then the path should travel clockwise. In order to determine whether a path is correctly oriented the angles and sense of rotation need to be computed for all circular coedges which occur in the loop between the on and off-edges.

3.2 Closed Cycle Searching

The result of the path generation phase of the algorithm is a *directed graph*. In the next stage the recognition algorithm proceeds by taking a list of all the paths which have been created and searching for all closed cycles in the graph. The cycle data structure contains a list which is empty at construction but which has paths added to it as the search proceeds. The search is initialised by the construction of a cycle, referred to here as the *present cycle,* and the selection of an arbitrary path. The path is added to the present cycle and a succeeding path is sought. A succeeding path is one which has an on-edge identical to the off-edge of the last path in the present cycle. This simple process of accretion is continued until one of two possible situations occur.

Firstly, more than one possible succeeding path may exist, in this case copies of the present cycle are made and the possible succeeding paths are added, one to each cycle. The copies of the present cycle are placed in a buffer to be completed later and the search is continued using the present cycle.

The second situation which may occur is that the goal of completing a closed cycle is achieved. The closing edge, i.e. the off-edge of the last path, has already appeared as the on-edge of a previous path in the cycle. The search may not have commenced with a path belonging to the cycle, in which case those paths prior to the first appearance of the closing edge are removed from the cycle.

If the buffer is not empty, then one of the incomplete cycles is set to be the present cycle and the search continues on that cycle. If the buffer is empty then the process is repeated starting with any path which has not yet been followed in the search. Such paths would form part of a disjoint subgraph.

A disadvantage of this refined algorithm search methodology is that by generating the paths *prior* to the search, in order to find the possible next paths in a cycle, all of the paths created over the entire model have to be checked rather than those which cross the subsequent face. This problem is heavily outweighed however by the absence of any geometric interrogations during the search, and in any case, is overcome by simply adding a list of next paths as an attribute for each edge at the path generation stage.

Next, duplicate cycles and self-intersecting cycles must be eliminated. Duplicate cycles are found quite frequently when the initial path is not actually part of the cycle and the cycle is reached by *different* routes. Each pair of cycles is checked for equality. Two cycles are equal, by definition, if they contain the same paths in the same sequence.

At this stage a list of closed cycles has been obtained and it is now required to *classify* these as either depressions or protrusions. The flow-chart for this process appears in Figure 5.

The closed cycles created by the algorithm must either bound a depression

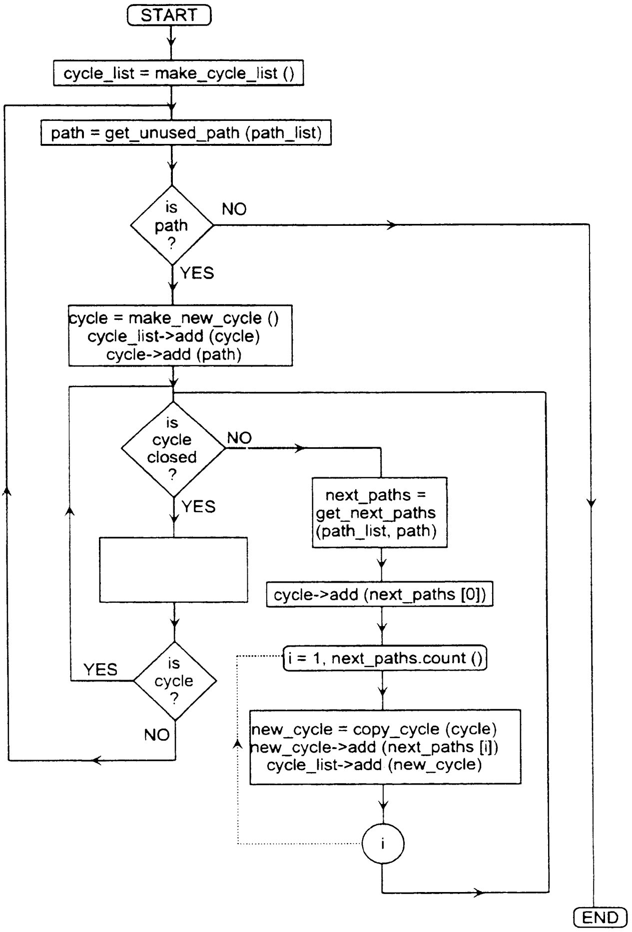

Figure 5. Flow Chart of Feature Cycle Creation

The closed cycles created by the algorithm must either bound a depression feature or a protrusion feature on the component. A cycle with the normal of the face traversed by its first path being directed out from the profile is a protrusion feature, whilst if the normal is directed in from the profile the cycle bounds a depression feature. The cycles are thus classified as either depression or protrusion cycles

By this method, all feature geometry for any solid model is created generically from the component topology. There is no requirement for any form of features library.

4. Data for Co-ordinate Measuring Machine

The geometric feature data stored as a result of the feature recognition is held as an ACIS file in ASCII format. A GUI has been designed which allows visual interaction for data transfer. Alternatively the data can be transferred automatically to the computer controlled processing unit of a CMM. The topological information contained within the data has a high degree of geometric accuracy ($\approx 1 \times 10^{-7}$) which allows for complex comparative inspection against pre-defined limits or tolerances. The allowable tolerance information can be attached to the data file as an *attribute*.

Due to the nature of the feature information generated it becomes a simple matter to specify, for example, a CMM geometry check on a particular component and to compare only against holes or some other specific features/areas of the part. Topological accuracy can also be compared with ease, e.g. *roundness* of a hole or *straightness* of an edge.

It is also possible to *reverse* the process. Rather than comparing the pre-constructed solid model features against a component, the component can be *mapped* by a CMM probe and the solid model constructed. The resulting model can then be interrogated by the feature recognition system. This method relies, naturally, upon a sufficient number of discrete probe co-ordinates to build an accurate *description* of the model.

Conclusions

A generic feature recognition algorithm has been developed which is capable of interrogating the geometry of CAD models of components and identifying the features without the need for a pre-defined features library.

Tolerancing information can be attached to the feature recognition output and the results compared with manufactured components when measured on a CMM.

Small scale shape variations between the solid model and actual components can be identified (e.g. roundness of holes, straightness of edges).

A robust and operational feature recognition user interface has been designed and prototyped. This can be used to save the data for transfer to CMM processor units. Alternatively, the data can be directly transferred by software integration.

A number of CMM manufacturing and end-user organisations have been involved in discussions over the use of this feature recognition architecture and its interfacing with CMM technology. These are set to continue and should result in ongoing research towards a commercially viable product.

References

1. Kyprianou L. K., *Shape Classification in Computer-Aided Design*, PhD Thesis, Christ's College, University of Cambridge, 1980.
2. Kim Y. S., Roe K. D., Conversions in form feature recognition using complex decomposition. Proc. 1992 ASME International Computers in Engineering Conference, Aug 2-6, San Francisco, CA, pp.233-244, 1992.
3. Sakurai H., Gossard D., Recognizing shape features in solid models. *IEEE Computer Graphics and Application*, September 1990, Vol.10, No.5, pp.22-32, 1990.
4. Sakurai H., Chin C.W., Definition and recognition of volume features for process planning, *Advances in Feature Based Manufacturing*, Elsevier/North Holland, 1994.
5. DeFloriani L., Feature Extraction from Boundary Models of Three-Dimensional Objects, *IEEE Transactions on Pattern and Machine Intelligence*, Vol.11, No.8, August 1989, pp.785-798, 1989.
6. Sormaz N., Clark D.E.R., Corney J., Tuttle J. R., A Feature Recognition Algorithm for NC-Machining. *Proceedings of The 5th IFIP Workshop*, WG5.2, Workshop on Geometric Modeling in CAD, May 1996, Airlee, Virginia, 1996.
7. Corney J., Clark D.E.R., Efficient Face-Based Feature Recognition, *Proc Second Symposium on Solid Modeling and Applications*, ACM Press 1993, pp.313-322, 1993.
8. Tuttle R., Sormaz N., Clark D.E.R., Corney J.R., Integrating Solids-Based Machining with Feature Recognition. *Proc 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, August 18-22, 1996, Irvine, CA, 1996.
9. PART White Paper, ICME Technologies, 1994

10. Risbud A., Automatic Feature Recognition, *Spatial Relations*, Vol 12, October 1995, Spatial Technology Inc, Boulder, Colorado, pp 3-4, 1995.

11. Roy, U., Computational Methodologies for Evaluating Form and Positional Tolerances in a Computer-Integrated Manufacturing System. *International Journal of Advanced Manufacturing Technology*, Vol.10, No.2, pp.110-117, 1995.

12. Pahk H.J., Jung M.Y., Hwang S.W., Kim Y.H., Hong Y.S., Kim S.G., Integrated Precision Inspection System for Manufacturing of Molds Having CAD Defined Features. *International Journal of Advanced Manufacturing Technology*, Vol.10, No.3, pp.198-207, 1995.

13. Zhang Y.F., Nee A.Y.C., Fuh J.Y.H., Neo K.S., Loy H.K., A Neural-Network Approach to Determining Optimal Inspection Sampling Size for CMM. *Computer Integrated Manufacturing Systems*, Vol.9, No.3, pp.161-169, 1996.

14. Merat F.L., Radack G.M., Automatic Inspection Planning Within a Feature-Based CAD System. *Robotics And Computer-Integrated Manufacturing*, Vol.9, No.1, pp.61-69, 1992.

15. Ge Q., ChenB., Smith P., Menq C.H., Tolerance Specification and Comparative-Analysis for Computer-Integrated Dimensional Inspection *International Journal of Production Research*, Vol.30, No.9, pp.2173-2197, 1992.

---

[a] ACIS is the copyright of Spatial Technology Inc., Boulder, USA.