

Advanced Feature Extraction and Recognition from Boundary Representation Models

McCormack, A.D., Department of Mechanical Engineering, Monash University
Ibrahim, R.N, Department of Mechanical Engineering, Monash University

Abstract

The function modeler presented in this paper was designed to extract product features for automatic process selection in applications most suited to vertical milling. This modeler works on principles of Boundary Representation (B-Rep), specifically Euler formula and Attributed Adjacency Graphs, and Constructive/De-Constructive Solid Geometry (CSG) modelling. Variations of the Adjacency Matrix and unique feature identification principles allow for the inclusion of interacting features. The product CAD data that is used within these principles is obtained from a neutral STEP file format, ensuring its compatibility amongst existing CAD programs. A recursive checking method, utilising volume decomposition of a CSG model, is employed to ensure that a valid product model is developed.

Keywords: Feature Extraction, Boundary Representation, Attributed Adjacency, Process Planning

Introduction

To enable automatic selection of process plans for an individual product, a Computer Aided Process Planning (CAPP) system must be able to extract detailed information from the product description. These components include geometry, dimensions, tolerances and surface conditions. Included in CAPP systems is the solid modeller, which consists of an interface for the user to input shape information, data structures to represent the CAD file, and an algorithm to generate the shape information for the specified application. The data representation scheme is an integral part of the solid modeller as it can be affected by the product model application. There are generally two recognised models (1); boundary representation (B-REP) and constructive solid geometry (CSG) (3,6).

The process selection algorithm being presented has been developed from a generative point of view, meaning that it creates a new process plan for each product. This algorithm seeks to create a solid model based on both previously mentioned representation techniques to utilise the advantages of both, while simultaneously compensating for their failings. It is important to note that the algorithm does not create two *product* models. A solid model created under a boundary representation model can be easily recognised as unique if its geometric data and topological relationships are identical. Therefore, it is an excellent way of comparing models, and aids in the location of a part surface, which is essential when dealing with complex or interacting features. The secondary CSG solid model is merely a way of verifying the proposed process plan.

Modified Attributed Adjacency

This initial product model that the algorithm creates is in the B-Rep form. It needs to read the topological and geometric data of the product from the CAD file and create a simple array for storing the data. As it is not possible for two different shaped objects to have the same topological relationships and geometry data, this is a very simple way of interpreting product data. When examining a product representation in B-Rep form a search can be done through the CAD file, we can compare the various topological matrices and geometric data to a database containing features typical to the application and machining environment. This simple method for storing the relevant data regarding a part allows for a very simple method of including new features into the database.

Attributed Adjacency is product representation tool often found in the form of a graph (6,7), or matrix structure. The face-adjacency and relationships are stored (1,2,4) and create what is often referred to as Face-Edge graph (or Attributed Adjacency graph), where a node will represent the face and the corresponding edges for the links. However, information regarding the type of relationship and a face attribute are also of great importance when feature recognition is required. Often the face relationship (edge) is given an *attribute* of 0, 1 or 9, which can represent data such as convex or concave relationships. Boundary representation can be much more useful if more detail can be extracted from the adjacency matrix. It is no longer sufficient to classify only edges in the matrix based on three attributes.

The algorithm being presented in this paper utilises a variant on the Attributed Adjacency Graph by further classifying the attribute of the edge and an attribute that characterises each face. This is possible by utilising the leading diagonal for face attributes and by extending the matrix into the third dimension to accommodate for the face data. Figure 1 displays an adjacency matrix in its basic form with element 8,8 in the matrix indicating that face 8 is of concave-curved type whereas all others are planar.

Face attributes do not increase the complexity of an Attributed Adjacency a great deal and this is an important part of this research. The ability to update a features database inside any feature extraction/identification framework is very important. Often of equal importance is the ability for people, other than the designer, to maintain the feature database and update it as they see fit. Including a new feature that essentially only consists of a simple arc-node graph, or matrix, is a task that can be accomplished by any person with basic education in product representation and computer aided drawing.

The additional edge data, which is not present in Figure 1, may include noting that the interaction is between a cylindrical opening and a planar face, or a concave curved surface and a smooth plane. In the case of two smooth plane faces forming an edge, either concave or convex, the new data that would be included in the matrix would be the angle that the two faces forming in constructing the edge. This type of data will prove extremely useful when identifying features. An example would be the ability to immediately distinguish between a square slot, a dovetail slot, and a V slot. All of these slots would have otherwise been recognised as a sequence of three planar faces with concave relationships.

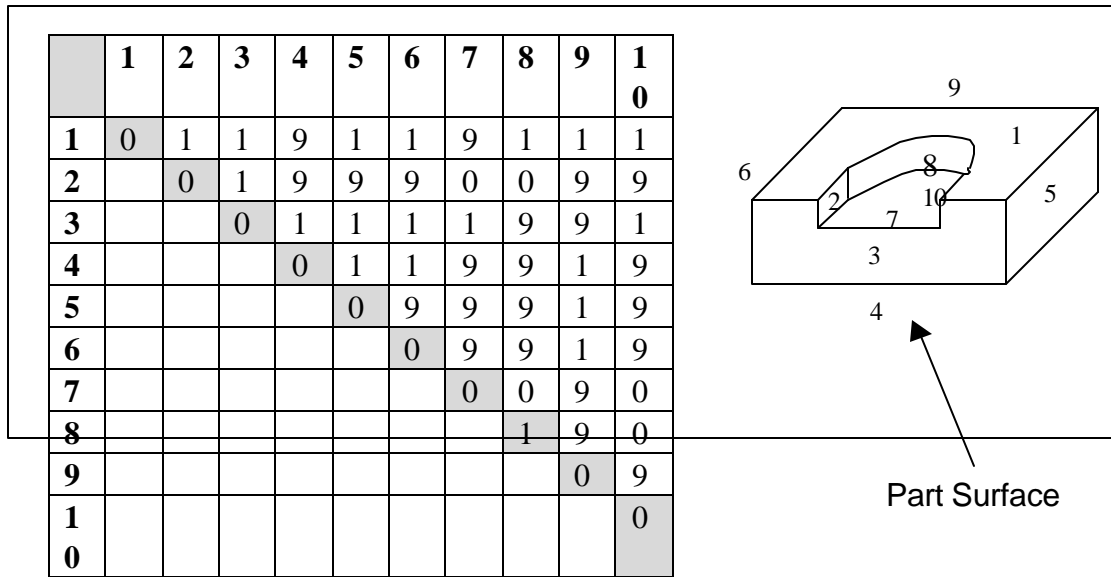


Figure 1.

Feature Extraction

Joshi and Chang (8) developed a heuristic for extracting sub-graphs/matrices from the Attributed Adjacency Matrix that is based on the statement: a face that is adjacent to all its neighbouring faces with a convex angle does not form part of a feature. What this means is that the end of a depression feature is determined by a convex edge, and a protrusion ends via a concave edge. This allows the features specific to the application to be extracted so that each individual machining operation can be identified. This is simplified by using the Euler equation (3) to determine whether the current feature is a passage or non-passage feature. Identification of a concave relationship should not end the search criteria. For a feature extraction system to be classified as robust, the search criteria should only be initiated by heuristics, not controlled. So much more information is available during this feature identification stage that can make process selection and path selection a very simple operation.

Upon identification of a concave edge formed by two faces, the whole row/column should be scanned to count the total number of concave edges that one face has associated with it. The search criteria will vary dependent on the type of feature i.e. corner, depression, protrusion. The faces forming the concave relationship should not be the last ones noted as has been the case in previous work (6,8). Those faces, which bound the individual feature, faces #3 and #1 in Figure 1, are of equal importance as they can directly affect process precedence. An example of the typical search criteria procedure for the presented algorithm will follow for depression style features such as slots or pockets. The basis of identification for a depression feature is two or more concave edges in one row or column, which then classifies it as the *part* surface. A similar concept can be found in (9) that deals with solids. This part

face is always the bottom face of a slot, blind hole or pocket and will most likely be the plane limiting any cutting.

The face labeled part surface in Figure 1, face #7, will have concave relationships formed with the surrounding faces. The total number of surrounding faces is not yet required; it is sufficient at this stage to know that it is forming a feature. The procedure that this algorithm employs is demonstrated as follows:

```
Part surface identified
  Call internal_ring_check
  If not hybrid_depression
    Return (all bounding faces)
  Else Call establish_precedence
Call connecting_convex_edge_map
  call features_database
  if open_feature
    determine approach direction
    compare feature_boundary_faces to stock_boundary_faces
    if corner
      return (boundary edges)
    else return (approach)
  call contour
  return (cutter_path)
```

Passage features are given a *genus* rating (2) to determining the level of feature complexity. The genus is other criteria with which the Euler equation must be compared to when dealing with non-simple objects, such as spheres or cylindrical through holes. Interacting features, such as two slots running perpendicular to each other, have often been separated in the past (8). But through the identification of the part surface the algorithm can recognise multi-faceted slots by mapping all faces, as performed in the procedure above that make a convex edge with the current part surface. This is why the external loop of a part face in question is extremely important. Instead of splitting a slot, that would otherwise have been considered interacting, into simple solids to remove, we map a contour/pocket that is machined out using the external loop of the part face and the process selection for all identified features is now possible.

Algorithm for Process Selection

Once the individual features sub-graphs have all been identified, they must be indexed based on complexity. An obvious example would be that a counter-sunk thread could not be tapped until an initial basic drilling operation was performed. The selection of each machining process for each individual feature will be made from data available in a database that is machine specific. This process will entail pattern-matching techniques between the individual feature sub-graphs/matrices and the process/feature database.

Once all features have been indexed, the dimensions, symmetry and orientation of each individual feature are examined simultaneously and compared to the current tool data to ensure machining compatibility. The accessibility should be viewed as the starting point for the individual feature's process path and will update the sequencing of each process. The determination of the approach direction was investigated by Aldakhilallah and Ramesh (4). They proposed that a system should develop a set of all the directions from which a tool can approach a face. For example $\{+x,-x,+y,-y,+z,-z\}$ would be a set possible for most three-dimensional products. From this set, the system needs to interpret a set of *feasible* directions from which a face can be machined. The data that is used in this step has originally been extract from the adjacency matrix and stored as the convex edge mapping of each part surface that is identified.

The machining data is now obtained and a product model is developed based on raw stock data and material removal of the current tool. A variant on the constructive solid geometry technique is utilised to form a trial of the product. This process is known as volume decomposition and is an excellent way of creating a secondary solid model as it refers to the volume removed by each process as an individual 3D solid.

The material that is removed via each process is linked with a specific 3D primitive. The dimensions are determined from the tool database and the original B-Rep solid model. Each solid also contains a predetermined reference point. This reference point is then positioned according to data regarding the current feature, which is obtained from the B-Rep model. The orientation of the solid has been calculated from the tool approach direction. Each solid is then subtracted from a user defined stock piece. This new CSG model has now essentially been created by an internal "design by features" function.

Upon *removal* of each three-dimensional solid, a check is performed on the current feature and it is converted into boundary representation model. What this means is that the algorithm can compare the topology matrix and geometry data of the newly removed feature and compare it to the corresponding feature in the original solid model. If the feature that is being proposed does not match the required feature, the algorithm loops back and examines other feasible data which were earlier ignored for others. If a problem arises in feature identification, this secondary CSG solid model can be quite useful for user prompting as this is an area which B-Rep often lacks (6).

Once all processes have been synthesised on this trial the geometric and topological data of the completed model are compared to the actual product model and the machining database to give an approximation of the final tolerance on the model.

Conclusion

The proposed algorithm developed using B-Rep and CSG techniques was used to determine machining processes for different products. In the algorithm, these techniques were employed to utilise the advantages of both in that the complexity of the product being modeled can increase and the time taken to recognise the feature can be reduced. This ability is derived from the modified Attributed

Adjacency Graph/Matrix and a unique method for dealing with complex or interacting features is explained. With a simple method for storing data regarding product features, identification of the *part* surface, and related machining processes, the process sequencing can be performed at a quicker speed provided the algorithm contains vigorous search and extraction procedures.

References

1. Willem F. Bronsvort and Frederick W. Jansen, Feature modelling and conversion - Key concepts to concurrent engineering, *Computers in Industry*, 21 (1993) 61-86, Elsevier
2. U. Rembold, B.O. Nnaji, A. Storr, *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, 1994
3. Michael E. Mortenson, *Geometric Modelling*, John Wiley & Sons, 1996
4. K. A. Aldakhilallah, R. Ramesh, An integrated framework for automated process planning: design and analysis, *International Journal of Production Research* 1998, Vol. 36, No. 4, 939-956
5. C.A. McMahon, D. R. Cox, J. H. Sims Williams, J. A. Scott, Representation and reasoning in computer aided process planning, *Proceedings from the Institute of Mechanical Engineers* Vol. 211 Part B
6. Nanua Singh, *Systems Approach to Computer-Integrated Design and Manufacturing*, John Wiley & Sons, Inc., 1996
7. C. Zhang, K.W. Chan, Y. H. Chen, A Method for Recognising Feature Interactions and Components within the Interactions, *International Journal of Advanced Manufacturing Technology* (1997) 13:713-722
8. S Joshi and T C Chang, Graph-based heuristics for recognition of machined features from a 3D solid model, *Computer-Aided Design* (1988) Volume 20 Number 2 March.
9. B.K. Choi, M.M. Barash and D.C. Anderson, Automatic recognition of machined surfaces from a 3D solid model, *Computer-Aided Design (1984)* Volume 16 Number 2 March

Title: Advanced Feature Extraction And Recognition From Boundary Representation Models

McCormack, A.D., Department of Mechanical Engineering, Monash University Department of Mechanical Engineering, Monash University, Caulfield 3145
Victoria, Australia

Email: raafat.ibrahim@eng.monash.edu.au

Andrew McCormack graduated from Monash University with the Bachelor of Engineering (Industrial & Computing) with honours. He is currently undertaking a Masters degree in the field of feature extraction and process planning

Ibrahim, R.N, Department of Mechanical Engineering, Monash University

Dr Raafat Ibrahim is a senior lecturer at Monash University in the Mechanical & Industrial Engineering Department and holds the following credentials: PhD, MEng, BSci(Hons), MIEAUST, CPEng