CHAPTER 12

Backward form feature recognition and removal for an automatic CNC-programming system - BCAM

Georg K. Christensen, Ole Bech Mogensen
Control Engineering Institute,
The Technical University of Denmark, 2800 Lyngby

A new concept for sequential form feature identification and elimination is presented. A set of domain-specific feature techniques are developed based on input from neutral formats of CAD-files with no explicit feature information. The method takes its inspiration from traditional process planning using a backward process planning concept. Various aspects of feature related CNC-path generation is treated and some physical test parts are presented. For the selected subset of relatively simple, but widely used geometry's, the BCAM system presents an effective method for automatic generation of CNC-code suitable for a 3D milling machine. Future enhancements and developments are discussed.

12.1. INTRODUCTION

With the widespread use of CAD systems in mechanical engineering industry, the interest in interfaces between CAD and CAM is increasing. One of the new concepts in the CAD/CAM integration is the use of 'features'. The following definition of a feature, presented by Shah [SHA 88], is very broad and can therefore be applied to a large number of points of interest:

Recurring patterns of information related to a part's description

In this context we shall restrict the concept to the smaller one 'form feature', which can be defined as follows:

> A geometrical entity, typically represented by a set of faces, which by its characteristic shape offers itself for various forms of chip removal processes

From the above definition, a form feature can be thought of as the geometrical representation of the removal volume in a chip removal process.

The development of an automatic CAD/CAM interface with the use of features has generally been proposed within the following two main philosophies [SAK 90]

R. Soenen et al. (eds.), *Advanced CAD/CAM Systems*

and [SHA 88]:

1 Design the part with a feature-based modelling system and thereby incorporate manufacturing information in the model at the design stage. The manufacturing information is preserved and transferred to the manufacturing stage.
2 Design the part with a solid modelling system and separate the design stage from the manufacturing stage . This means that no manufacturing specific information is explicitly present in the solid model. The manufacturing information is retrieved from the geometrical representation of the component alone.

In our work the second approach is followed using CAD information based on the neutral CAD-file format called NIRO-STEP [KRO 91]. This CAD file is slightly pre-processed into a format, that contains more explicit geometrical information and is then passed to the automatic CNC programming system, called BCAM. Since BCAM is an independent program, based on input from a neutral CAD file format, it will accept CAD models from various CAD systems independent of the internal data representation of the CAD system.

This concept presents some difficulties because none of the geometrical and topological operations available in the CAD system are available for the CNC programming system. This means that a set of geometrical and topological operations have to be implemented in the CNC programming system. This inconvenience has been overcome in the BCAM system, -not for general 3D shapes, -but for a widely used subclass of prismatic parts, capable of being manufactured on a 2_D milling machine (a machine with interpolation in the XY-plane, but without interpolation in the Z -axis).

12.2. STRUCTURE OF THE BOUNDARY REPRESENTATION (B-REP.) SOLID MODEL

As a prerequisite for the feature identification we use a B-Rep. solid model. A solid model has the fundamental characteristic, that it is possible to classify any point in world space in one of following three categories:

1 Inside the object
2 Outside the object
3 On the boundary of the object

The are two dominant types of solid models. One of these is the CSG model (Constructive Solid Geometry). This type of solid model holds no explicit information concerning the geometry of the part, but instead describes how to obtain the part from a series of addition and subtraction operations performed on geometrical entities. Since the CSG model does not directly describe the geometry of the resulting part, the CSG model is often called an implicit part description. Because there is no explicit geometrical information, this kind of model is not as attractive as the Boundary Representation model when it comes to feature identification and process planning. The B-Rep. model is an explicit geometrical description of the part. Basically a B-Rep. is defined by a set of faces with detailed information describing how the faces are connected. The general structure of a B-Rep. model is shown in fig. 1. This general B-Rep. model allows faces to have any geometrical appearance, but we have selected a subset called a B-Rep. polyhedron in which faces may only be planar. The structure of a polyhedron is shown in fig. 2. Since all faces are restricted to being planar and all edges linear, it is not necessary to have a geometrical definition of these entities.

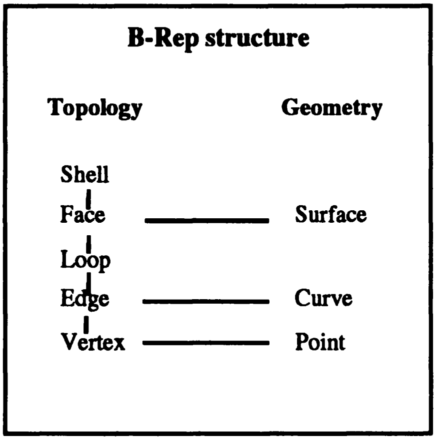

Figure 1. B-Rep structure

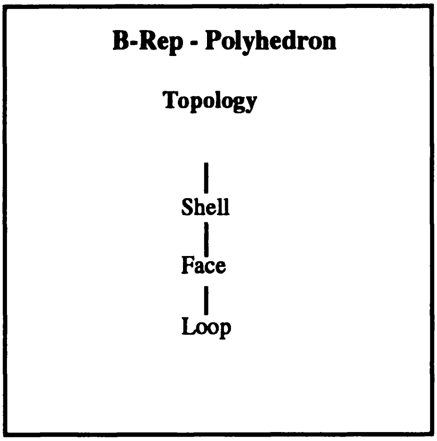

Figure 2. B-Rep polyhedron structure

Although the principles applied in our work are general, we have chosen to limit the investigations to polyhedron models of 2_D components. These parts are characterized by the fact, that the planar faces of the model have to be perpendicular to the XY-plane or parallel to the XY-plane..

12.3. ESSENTIAL MODEL CHARACTERISTICS

Some features of the B-Rep.model are essential for form feature identification. In our simplified polyhedron model these are related to face orientations, loop definitions and orientation rules. One important rule is, that every edge is represented in two and only two loops, meaning that edges originate from the intersection of two faces. This is illustrated in fig. 3.

Adjacent faces can thus be identified because they refer to the same edge. In this way all adjacent faces to a face can be determined by traversing the edge-list's for every loop in the specified face. Moreover the loop orientation on every face is defined, so an area vector, calculated from Green's theorem, points from solid to void. All inner-loops have the opposite orientation of the outer loop (see fig. 3), which means that they do not provide the necessary information for determining whether they mark the beginning of a protrusion or a depression. The presence of depressions and protrusions must be determined from the angle between the face normals of the two adjacent faces. If the angle is concave (less than 180 degrees), the feature is a protrusion, and if the angle is convex (larger than 180 degrees), the feature is a depression.

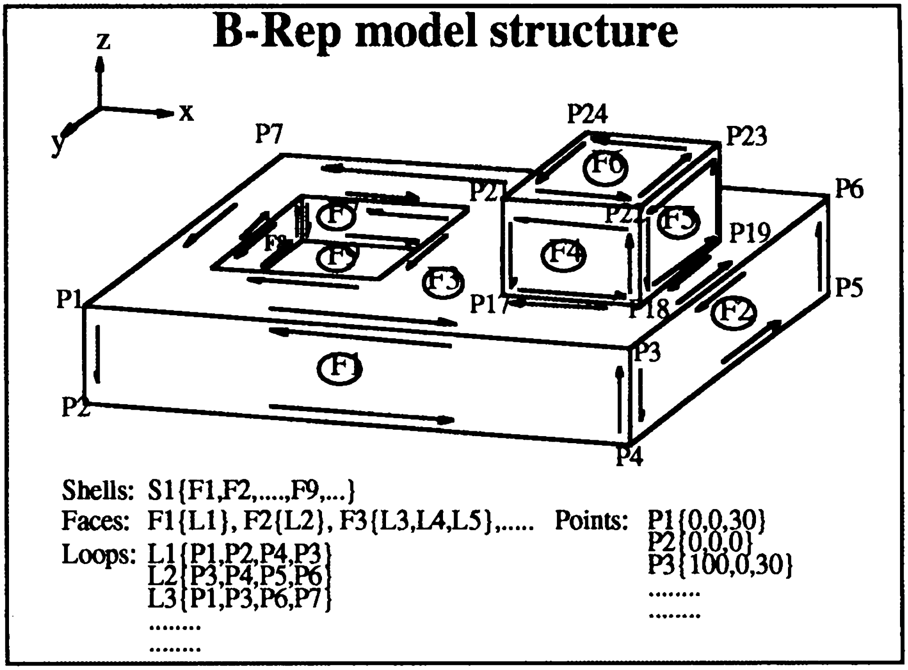

Figure 3. B-Rep structure

12.4. STRUCTURE OF THE BCAM SYSTEM

The BCAM system shown in fig. 4 is based on feature recognition in a neutral format B-Rep. polyhedron file. The models we use are designed with the CATIA system [CAT 90] and pre-processed with an in-house developed pre-processor into the NIRO-STEP polyhedron format [KRO 91].

Initially BCAM performs slight post processing where some implicit geometrical information is calculated and stored explicitly. This post processing step serves the purpose of saving computational time later during the time consuming feature identification process.

The Form Feature Recogniser decomposes the part into form features, that can be manufactured in a milling operation known as pocket milling. The data output of the FFR module is called a feature model, organized as a list of features according to the depth of appearance in the part.

The feature model data is passed as input to a CNC-program generator (see fig. 4). This program is able to convert the feature model into CNC code. The output of this module is a complete CNC program in G- and M- codes, which can be downloaded to a 2_ axis CNC mill. Optionally the verification module can be used to simulate and thereby verify the CNC program visually before download and execution on the CNC machine.

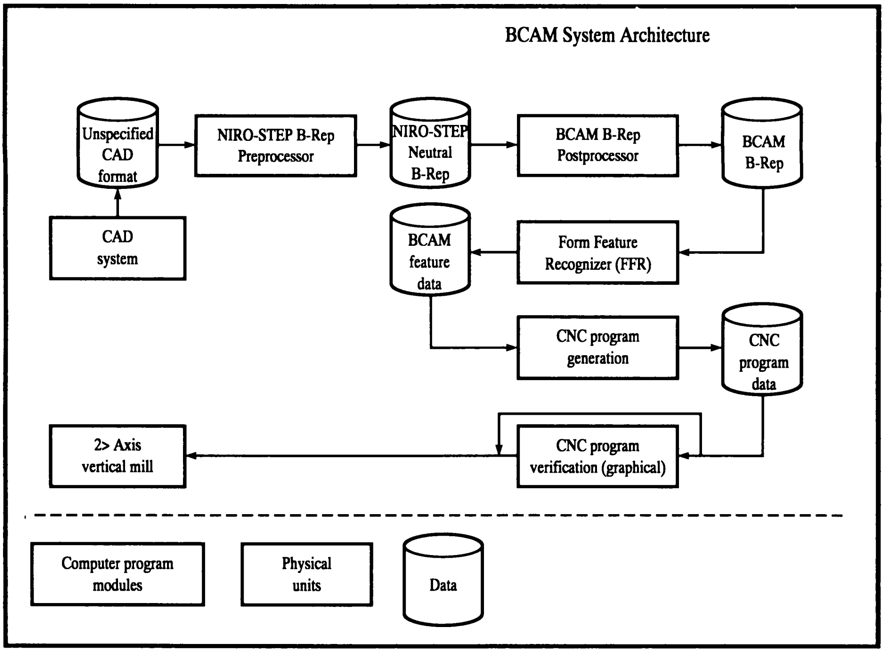

Figure 4. BCAM system architecture

12.5. FEATURE CLASSIFICATION IN BCAM

The feature classification scheme is a modified version of Gindy's feature classification scheme [GIN 89], which is particularly suitable for milling operations. The feature classification used, primarily divides features into three categories: Depressions, Protrusions and Peripheral features (see fig. 5).

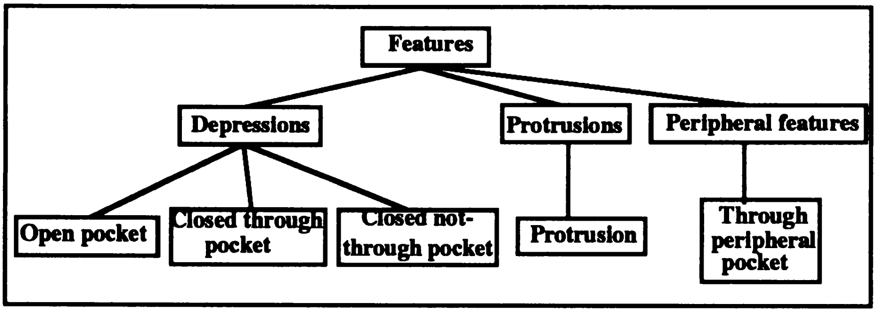

Figure 5. Feature classification scheme

Peripheral features are dealt with as a separate case. Considering a part to be manufactured from a rectangular block of material, the designer might have designed the part shown in fig. 6.

Viewing the part from above along the tool axis, the feature might be classified as a 'half hole' in the peripheral contour. These kind of boundary features are dealtwith as a special case. The boundary of the features describes the volume to be removed, but in this case it depends on the shape of the raw material block. It is therefore necessary to extract the feature boundary by comparing the part with a raw material block in order to determine the volume to be removed.

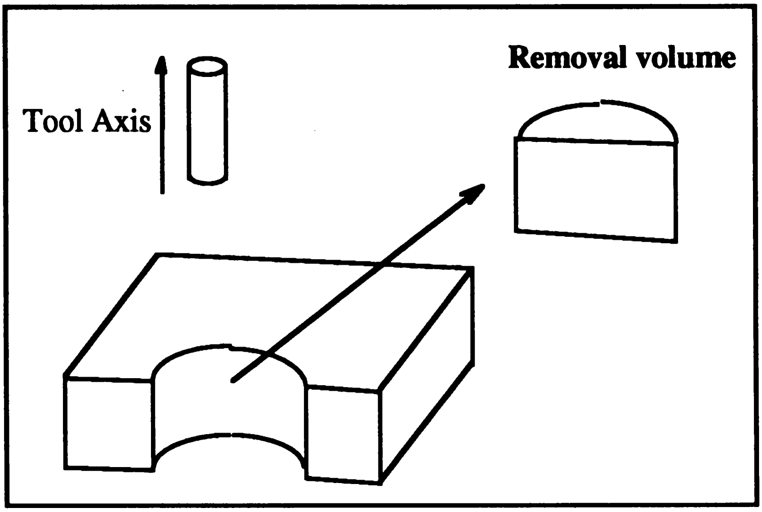

Figure 6. Peripheral feature

The reason for not initially distinguishing more feature variants is, that we prefer to handle feature identification in the most general way possible at this early stage. One of the reasons for this is, that we have considered it very time consuming to develop and implement many specific feature identification algorithms.

12.6. BACKWARD FEATURE RECOGNITION AND ELIMINATION

The feature identification algorithms are based on strict logical rules. Examples are the recognition rules for the open pocket shown in fig. 7.

The combined feature identification and feature elimination is closely related to backward process planning. When a feature is identified, it is removed from the CAD model analogous to performing a union operation between the part and the identified feature boundary. This process is more likely understood by process planners as 'filling material into the part' or reversing the process sequence.

According to the rules shown in fig. 7. it can be seen on fig. 8, that it is impossible to follow the second recognition and elimination sequence, because one of the side faces has more than 4 edges in the outer loop. In this particular case, only the first recognition and elimination sequence is possible.

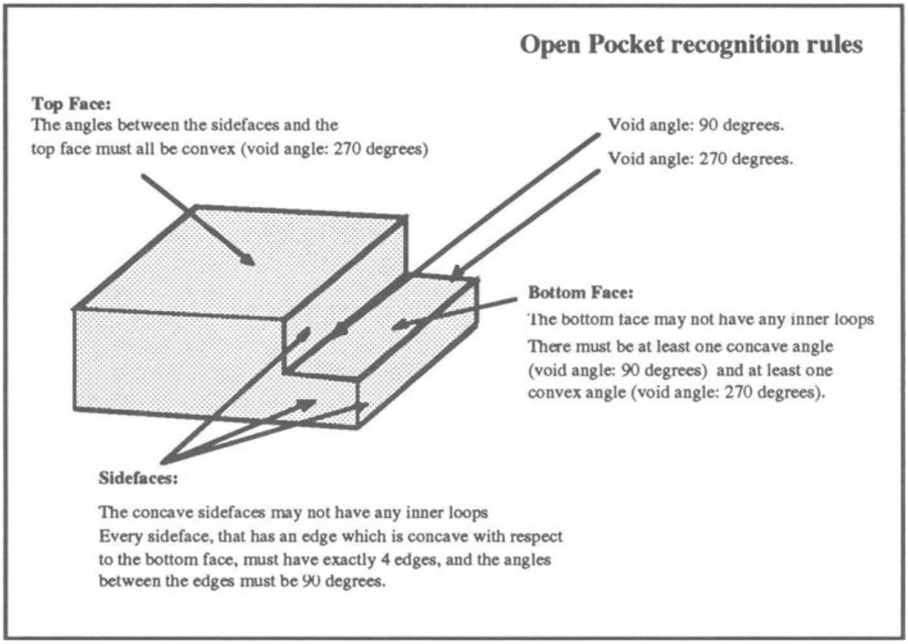

Figure 7. Open pocket recognition rules.

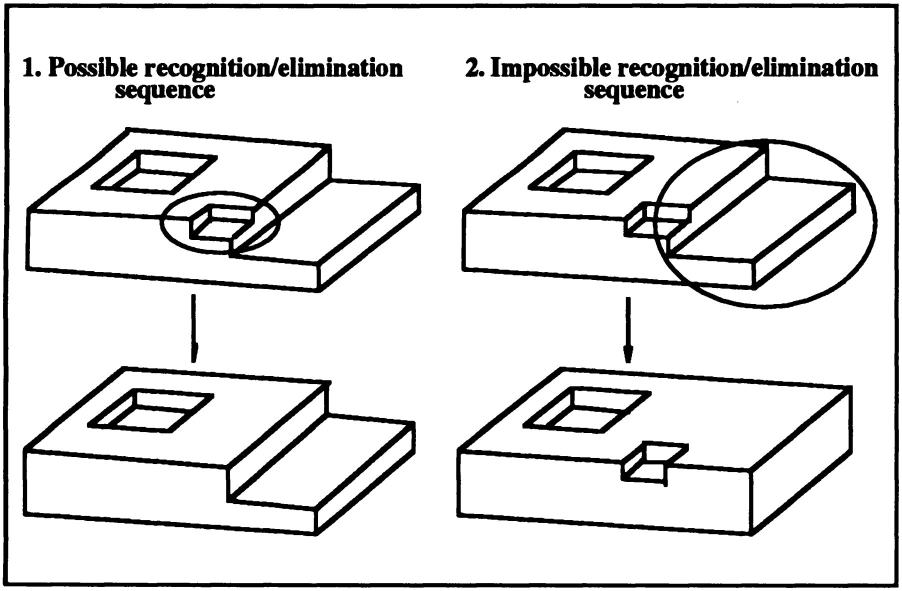

Figure 8. Openpocket elimination.

The order of recognition and elimination of features vary and depends on the actual part, which indicates, that an iterative method is required for building a complete feature model of the part. BCAM is therefore based on an iterative algorithm, that searches the part and tries to match a subset of the part as a feature according to the feature identification rules. When succeeding, the feature is removed from the part and stored in the feature model. The iterative method stops, when no more features can be removed from the part. At this point BCAM has calculated the necessary raw material block for the part manufacture. An example illustrating the whole iterative process is shown in fig. 9.

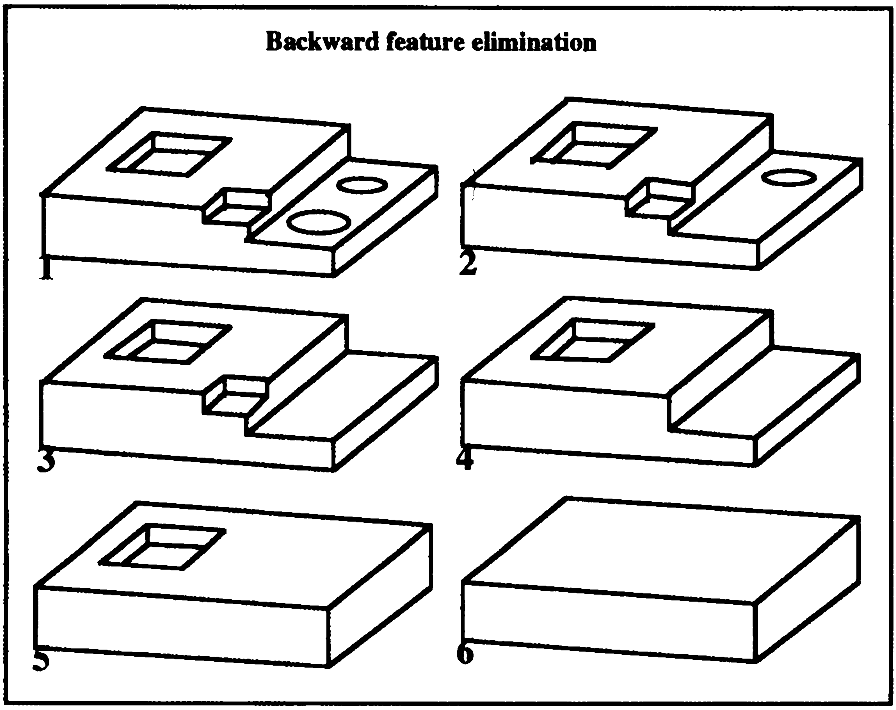
Backward feature elimination

Figure 9. Backward feature elimination

One of the major tasks in developing the feature identification and elimination algorithm lies in developing a general feature elimination algorithm. This algorithm must be able to perform a union operation between two solids and construct a valid boundary representation of the resulting volume. This operation is incorporated in solid modelling systems, but only a reduced version has been integrated in BCAM. The principles of reconstructing a valid topological and geometrical B-Rep. structure are documented [ARI 85]. However we judged it to be too time consuming to develop and implement. Therefore a simpler mechanism has been integrated in this first version of BCAM. The developed mechanism can revalidate the B-Rep. structure after translation of a face along the z-axis as shown in fig. 10. The algorithm can revalidate the part after any XY-plane face translation along the z-axis, which means that most pockets, independent of boundary complexity, can be eliminated from the part.

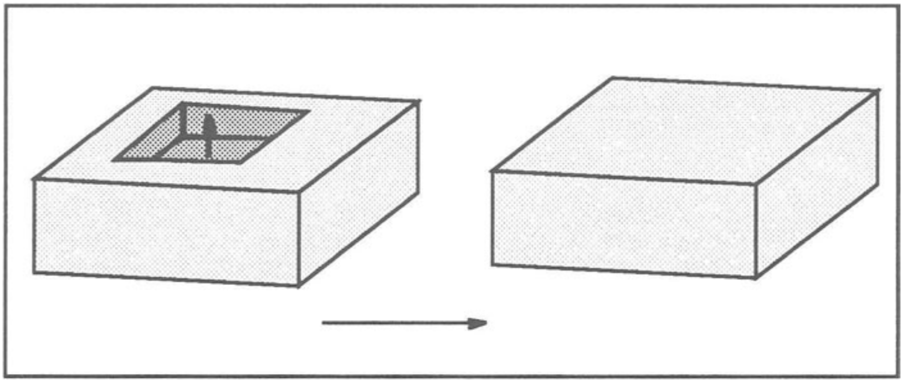

Figure 10. Face translation along z-axis

12.7. CONVERTING FORM FEATURES TO TOOL PATHS.

Considering open pockets and closed pockets, the tool path generation is now done feature by feature. At the present stage no effort is done to handle composite features. By composite features we mean features that by a more strict process evaluation system could be classified as intersecting features as shown by the example in fig. 11.

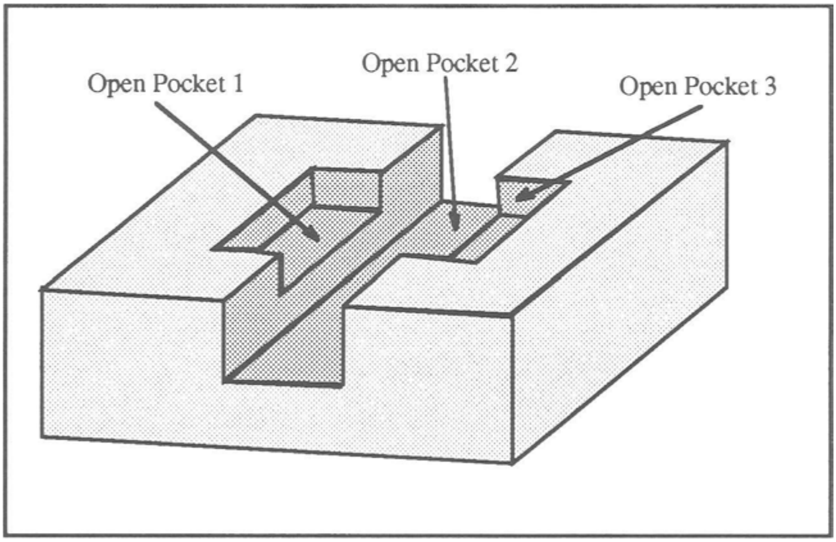

Figure 11. Composite features

For this kind of features BCAM reduces manufacturing to the problem of pocket machining, that is a quite well known problem [HOS 85],[TIL 84],[SUH 90] and [HEL 92]. This is done by first converting all open pockets to closed pockets. This conversion process is a matter of translating all convex pocket boundary edges _ tool diameter as shown in fig. 12. The edge translations are necessary in order to avoid incorrect cutting as shown in fig. 13. The algorithms of Suh and Lee, [SUH 90] was implemented for the milling of arbitrary polygonal pockets.

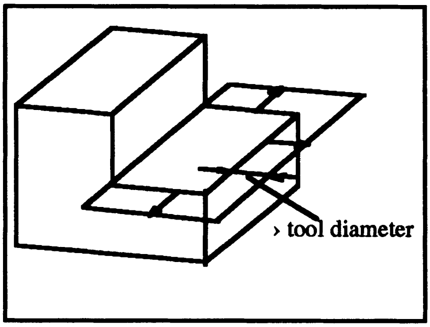

Figure 12. Pocket edge translation

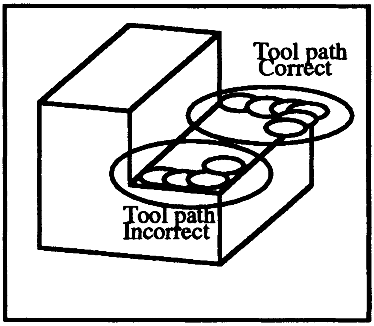

Figure 13. Done

The peripheral features are handled in a different way. First they have to be treated as open pockets, so they can be manufactured by pocket milling. Alternatively, depending on machining conditions, the peripheral features are not machined separately, but just removed by a milling operation following the outer contour.

12.8. TEST AND EVALUATION.

The BCAM system is implemented under the OS/2 ver. 2.0 operating system in the 'C' language and ran on a 25 MHz, 80486 processor. The system has been tested on several 2_ D models. An example is shown in fig. 14.

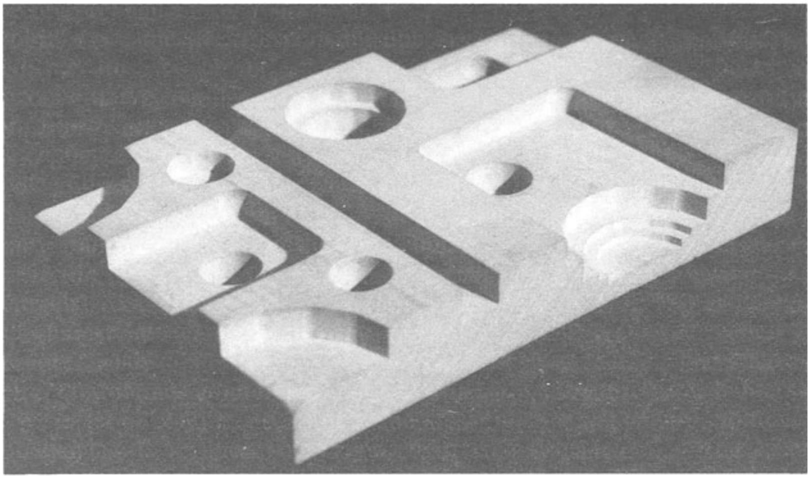

Fig. 14: A test case

The computer calculation time for this example is about 10 seconds. After visual verification the generated code was downloaded and run on an educational 3-axis milling machine (Triac, Denford Machine Tools).

The test showed that the usage of polygonal approximation of circles gave a very slow machining of these. One reason for this is, that the machine decelerates for each linear increment . This might not be a problem on CNC machines that contains an automatic corner override mode in the form of a G62-code. However it is felt that this was an area of future improvements of the system. To achieve the improvement it will be necessary to identify circular features as such or "glue" the linear segments together to form circular arcs in accordance with some error criteria.

12.9. FUTURE DEVELOPMENTS

Presently the feature identification algorithms are 'hard coded' into the BCAM system. A user accessible feature definition system should be developed, so that the feature database can be modified and extended by a super-user according to specific needs.

Another immediate improvement of the system would be to incorporate the placement of fixtures in the feature model and make the CNC program generator able to take the placement of these fixtures into account.

The system capabilities can also be extended by incorporating the capabilities of altering access directions for the identification of features. This would enable the identification of features that are only accessible for manufacture and identification from these directions.

Last the pocket milling algorithms should be further improved in order to be able to generate CNC code for arbitrary polygonal pockets with any number of islands.

The main challenge seems however to lie in the extension of the geometrical representations to true 3D capabilities and in a more sophisticated analysis of the feature model in order to identify processes for optimized manufacture using other processes in addition to CNC-milling.

12.10. CONCLUSIONS

This chapter has suggested and explored a new concept of backward feature identification and removal for automated generation of CNC-program. The necessary algorithms for feature identification , removal and for model restoration have been described. The geometry's used have been simple and it still remains to be seen how far this concept can be extended into handling of full 3D models. The system produced can however generate CNC-code for 2_D models fast and accurately using no further information from the CAD system than a B-rep. polyhedron solid model.

12.11. REFERENCES

[SHA 88] Shah, J. and Rogers, MT: "Expert form feature modelling shell.", Computer Aided Design, vol. 20, no. 9, nov. 1988.

[SAK 90] Sakurai, Hiroshi and Gossard, David C.: "Recognising shape features in Solid models", IEEE Computer graphics and applications, sep. 1990.

[SHA 88] Jami J. Shah, Mary T. Rogers: "Functional requirements and conceptual design of the Feature-Based Modelling System", Computer Aided Engineering Journal, Feb. 1988.

[KRO 91] Uri I. Kroszynski, Torben Sørensen and E. G. Schlechtendahl: "NIRO-STEP Specification for the transfer of Robotics Models", Control Engineering Institute, Technical University of Denmark, IFS Rap. nr. 91.90.

[CAT 90] CATIA Base, version 3, Interactive Functions Reference Manual, Publ. nr. 50-0098-03, IBM, 1990.

[ISO] ISO standard 10.303-STEP, PART 42,43 and 105.

[GIN 89] Gindy, N. N. Z. : "A hierarchical structure for form features", Int. Journ. of Production Research, vol. 27, no. 12, pp. 2089-2103, 1989

[ARI 85] Aristedes, A.; G. Requicha and Herbert B. Voelcker: "Boolean operations in Solid Modelling: Boundary evaluation and merging Algorithms", proceedings of the IEEE, vol. 73, nr. 1, jan. 1985.

[HOS 85] Josef Hoschek: "Offser curves in the plane", Computer Aided Design, vol. 17 no. 2 ,marts 1985.

[TIL 84] Wayne Tiller and Eric G. Hanson: "Offset of two-dimensional profiles", IEEE Computer Graph. & Appl. Vol. 4 No. 9 1984, pp. 36-46.

[SUH 90] Yong Seok Suh & Kunwoo Lee: " NC milling tool path generation for arbitrary pockets defined by sculptured surfaces", Computer Aided Design Vol. 22 No. 5 June 1990.

[HEL 92] M. Held: "On the computational Geometry of Pocket Machining", Lecture Notes in Computer Science, 500, Springer Verlag, 1992.