Int J Adv Manuf Technol (2002) 20:553–556
Ownership and Copyright

The International Journal of Advanced Manufacturing Technology

Feature Recognition Patterns for Form Features Using Boundary Representation Models

N. Ismail[1], N. Abu Bakar[2] and A. H. Juri[3]

[1]Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University Putra Malaysia, Serdang, Malaysia; [2]Business and Advanced Technology Centre (BATC), University Technology Malaysia, Kuala Lumpur, Malaysia; and [3]German-Malaysian Institute, Cheras, Kuala Lumpur, Malaysia

This paper present a new method for automatic form feature recognition from boundary representation of a solid model. The proposed method is called the edge boundary classification technique. This technique uses the geometry, topology, and spatial information of a solid model for feature recognition patterns. Some aspects of edge boundary classification are discussed for recognising non-intersecting and intersecting form features.

Keywords: Edge boundary classification; Feature recognition; Features

1. Introduction

Automated Feature recognition has been attempted using many methodologies for a wide range of applications [1, 2]. The methods for recognising feature depend on the geometric model used such as a solid model (constructive solid geometric (CSG) or boundary representation (B-rep)). A general review of the application of features in manufacturing can be found in Pratt [3] and Allada and Anand [4].

Most of the earlier works on feature recognition from B-rep solid models used only the geometric and topological information extracted from the geometric database. The techniques involve complex transformation of geometric and topological data into a different format before feature recognition is carried out. Because of the complexities of transformating data into a usable format, the techniques are usually limited to a specific class of either polyhedral or cylindrical type features. This work extends the use of the information for form feature recognition to include the spatial addressability information available in a solid model.

Correspondence and offprint requests to: Dr N. Ismail, Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University Putra Malaysia, 4300UPM, Serdang, Malaysia. E-mail: napsiah@eng.upm.edu.my

Spatial addressability information is a special characteristic of solid modellers where a point in space can be classified relative to the object model as to whether it is *inside*, *outside*, or *on* the object [5]. The use of spatial information simplifies the recognition process as no geometric and topological data transformation need be performed on the solid model. Only additional information (i.e. the spatial information) needs to be processed and made explicit.

2. Edge Boundary Classification

The database of a B-rep solid model contains both *geometry* and *topology* data. The geometric elements that define objects are the length of lines, the angle between the lines, the radius, and the centres of circles. Topology, on the other hand, is the connectivity and associativity of the object entities. The B-rep model of an object (B) can be defined as $B = (V, E, F)$ where V = {set of vertices}, E = {set of edges}, F = {set of faces}. The edge boundary classification (EBC) technique is based on several basic concepts developed from the B-rep data structure such as edge loop. This work concentrates only on planar face edge loops.

The classification of each edge in an edge loop for planar faces will result in a pattern that represents the spatial addressability characteristics of the edge loop of its outer boundary and inner boundary. The outer boundary (out_bound) and the inner boundary (in_bound) concept is shown in Fig. 1.

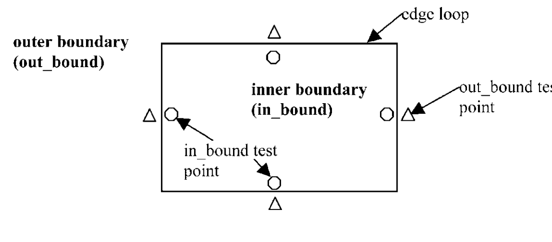

Fig. 1. Edge boundary classification concept.

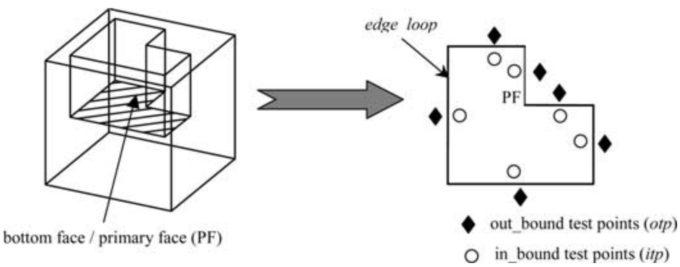

Fig. 2. A blind pocket.

Given an edge loop, an edge boundary classification pattern can be formed from the results of classifying the in_bound and out_bound test points of the edges in the loop with reference to the solid model. The result from the classification of the test points for an edge loop can be as follows [6]:

Inside the solid "*inobject*" denoted by the value 3.
On the solid "*onobject*" denoted by the value 2.
Off the solid "*offobject*" denoted by the value 1.

3. Feature Patterns

A feature is recognised if the recogniser can detect the sequence of the pattern of the primary face corresponding to feature pattern in the library. To illustrate the concept of EBC, a blind pocket feature and a through step are discussed in this section (Fig. 2). For a blind pocket feature, the bottom face can defined as the primary face. Analysis of the EBC test points for the primary face gives the following conclusions:

All in_bound points of the primary face edge loop are *onobject* (point value = 2).
All out_bound points of the primary face edge loop are *inobject* (point value = 3).

We can define the EBC pattern for an N-sided blind pocket to be:

in__bound = $(itp_1, itp_2, itp_3, itp_{N-1}, itp_N)$
out__bound = $(otp_1, otp_2, otp_3, otp_{N-1}, otp_N)$
where $itp_1 = itp_2 = itp_3 = itp_{N-1} = itp_N = 2;$
$otp_1 = otp_2 = otp_3 = otp_{N-1} = otp_N = 3;$
and $N >= 3.$

This EBC pattern can also be represented as shown in Table 1.

Table 1. EBC pattern for an N-sided blind pocket

Edge loop classification		Edges of PF e_1	e_2	:	e_{N-1}	e_N	Feature recognised
in_bound	Pattern 1	2	2	:	2	2	N-sided blind pocket
out_bound	Pattern 1	3	3	:	3	3	

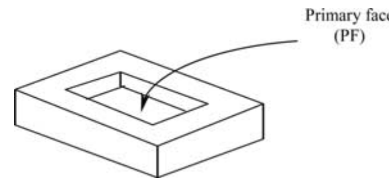

Fig. 3. A four-sided blind pocket.

Table 2. EBC pattern for a four-sided blind pocket

Edge loop classification		Edges of PF e_1	e_2	e_3	e_4	Feature recognised
in_bound	Pattern 1	2	2	2	2	Blind pocket
out_bound	Pattern 1	3	3	3	3	

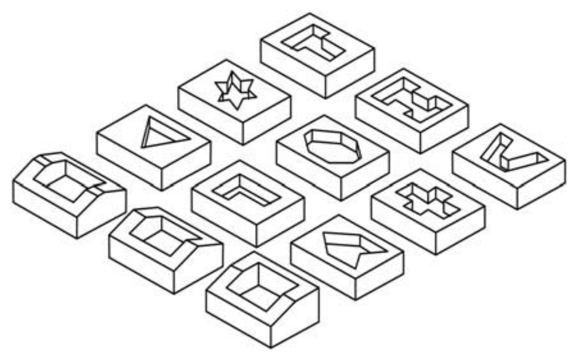

Fig. 4. Different instances of blind pockets.

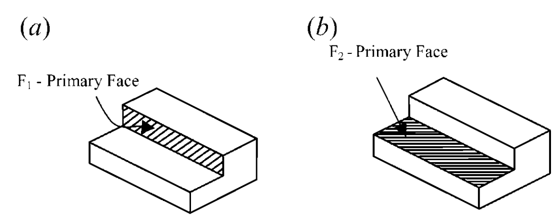

Fig. 5. Through step with two primary face possibilities.

For a four-sided blind pocket, as shown in Fig. 3, the in_bound and out_bound EBC pattern is shown in Table 2.

The EBC pattern for blind pockets is generic and applicable to various instances of blind pockets, some of which are shown in Fig. 4. The patterns defined do not differentiate between these unbounded varieties of pocket features. At the first level of form feature recognition, the technique is, however, able to recognise the feature type quickly, enabling further detailed geometric and topological data to be extracted from the geometric model.

A through step is defined as a feature which has not more than two faces connected together. For purposes of feature recognition using the EBC technique, either of the two faces can be used as a primary face. Figure 5 shows an example of a through step feature and the primary face possibilities. An

Table 3. EBC pattern for four-sided through step

Edge loop classification		Edges of PF e_1	e_2	e_3	e_4	Feature recognised
in_bound	Pattern 1	2	2	2	2	
	Pattern 1	1	1	3	1	Through step
out_bound	Pattern 2	1	1	1	3	
	Pattern 3	1	3	1	1	
	Pattern 4	3	1	1	1	

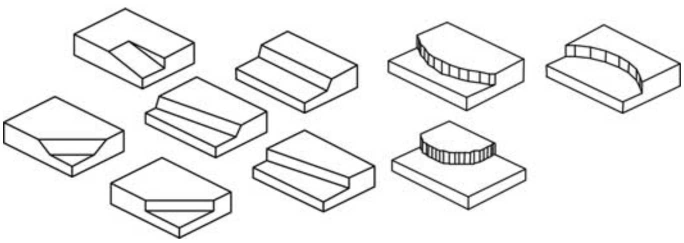

Fig. 6. Different types of step feature.

EBC analysis of the primary face shows that an edge loop will have the following properties:

The in_bound pattern is all *onobject* (value = 2).

For the out_bound pattern, all the test point value must be *offobject*, except that one and only one test point value must be *inobject*.

The EBC pattern of a through step feature is shown in Table 3. Figure 6 shows different examples of steps with linear and curved loops that are classified as through steps.

4. Framework of Recognising Features

Figure 7 shows a general overview of the experimental feature recognition system developed to test the technique. The AutoCAD solid modelling system [7] is used to provide the development tools required. It uses the constructive solid geometry (CSG) modelling technique to define solids using six basic primitives and also maintains a B-rep database of the solid model. Each solid model thus contains CSG information that maintains the structure and dimension of the solid and the B-rep information to describe the boundaries of the model. The input information to the experimental feature recognition system is the B-rep model and the output is form features such as pockets, slots, and steps. These recognised features can be used in computer aided-manufacturing (CAM) systems.

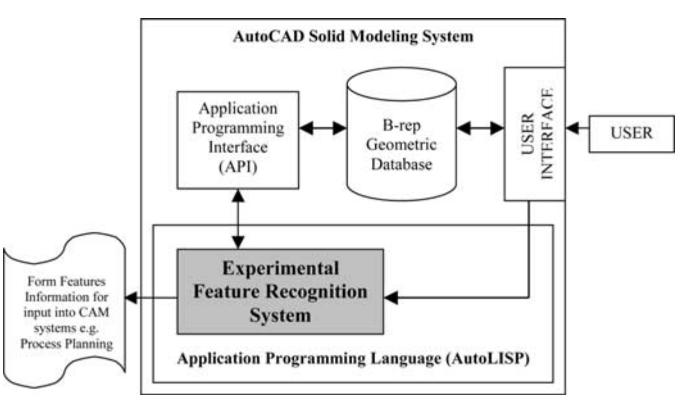

Fig. 7. Framework of recognising features.

A feature recognition system to recognise form features from a B-rep solid modelling system using the EBC techniques proposed must be able to:

1. Retrieve the associated entity data name of the current solid model being processed from the database.
2. Extract all relevant geometric and topological data and pre-process the information into a format suitable for use by the EBC algorithms.
3. Process geometric and topological data provided using the EBC technique to establish spatial addressability information and form EBC patterns.
4. Perform feature recognition by pattern matching and extraction of feature parameters from the geometric database.

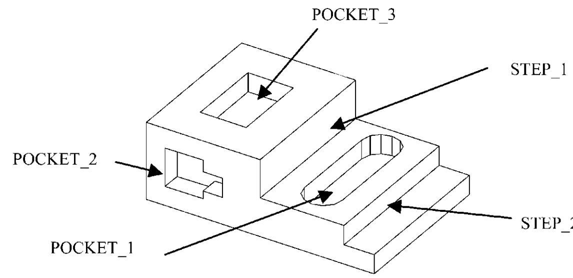

Fig. 8. Test part.

5. Results

To illustrate the capabilities of the recognition system proposed in this paper, a test part, as shown in Fig. 8, is used. The part consists of 25 faces including 2 semi-cyclinder faces and 72 edges with 4 circular edges. Table 4 shows the list of features recognised by the system, namely one 8-sided blind pocket (BLIND_POCKET2), two 4-sided blind pockets (BLIND_POCKET1 and BLIND_POCKET3) and two steps

Table 4. Feature recognised

Blind_Pocket_list"
(BLIND_POCKET1 11.0 (59.0 57.0 51.0 54.0) 13.0 (58.0 56.0 49.0 53.0) 10.0 9.0 7.0 8.0))
(BLIND_POCKET2 4.0 (43.0 42.0 41.0 34.0 31.0 40.0 27.0 35.0) 21.0 (83.0 32.0 38.0 45.0 46.0 63.0 37.0 28.0) (11.0 6.0 5.0 3.0 2.0 17.0 1.0 3.0))
(BLIND_POCKET3 15.0 (77.0 76.0 75.0 74.0) 23.0 (86.0 84.0 81.0 79.0) (18.0 19.0 16.0 17.0))
"Step_list"
(STEP1 13.0 (70.0 69.0 68.0 65.0) (14.0))
(STEP2 12.0 (67.0 65.0 66.0 60.0) (11.0))

(STEP1 and STEP2). As an example, the results for BLIND_-POCKET1 can be interpreted as follows: primary face-id 11.0 with associated edges, edge_id (59.0 57.0 51.0 54.0), entrance face-id 13 with associated edges, edge-id (58.0 56.0 49.0 53.0) and four wall faces adjacent to primary face, face-id (10.0 9.0 7.0 8.0). The face-id 7.0 and 10.0 are the semi-cylinder faces. For a step feature, the list can be read as the feature-id, primary face-id, primary edges, and adjacent wall face.

6. Discussion and Conclusion

The implementation of the proposed EBC technique in an experimental feature recognition system using the development tools provided by the AutoCAD modelling system has been presented. Despite its low cost, the AutoCAD system has all the tools needed for the development work.

The technique can be used to recognise generic types of polyhedral feature such as pockets, steps, blocks, slots, without being constrained by the edge types (concave or convex) and loop type (linear, circular, or hybrid). The potential ability of the EBC technique to handle interacting features is also considered good because of its ability to be unaffected by geometric and topological changes not affecting primary faces, so, no additional post-processing should be required for the EBC technique.

References

1. J. J. Shah and M. Mantyla, Parametric and Feature-based CAD/CAM, John Wiley, 1995.
2. J. Shah, M. Mantyla and D Nau, "Introduction to feature based manufacturing", in J. J. Shah, M. Mantyla and D. S. Nau, Advances in Feature Based Manufacturing, pp. 1–11, Elsevier, 1994.
3. M. J. Pratt, "Application of feature recognition in product life cycle", International Journal of Computer Integrated Manufacturing, 6 (1&2), pp. 13–19, 1993.
4. V. Allada and S. Anand, "Feature-based modelling approaches for integrated manufacturing: state-of-the-art survey and future research directions", International Journal of Computer Integrated Manufacturing. 8(6), pp. 411–440, 1995.
5. Ibrahim Zeid, CAD/CAM Theory and Practice, McGraw–Hill, 1991.
6. N. Ismail and N. Abu, "Automatic recognition of machined features using edge boundary classification approach", Proceeding of International Symposium of Artificial Intelligent and Real Time Control, pp. 372–376, Kuala Lumpur, 22–25 September 1997.
7. AutoCAD R12, AutoDesk Inc. Advanced Modeling Extension (AME), 1993.