

Development of an Intelligent Database System to Automate the Recognition of Machining Features from a Solid Model Using Graph Theory

Rachna Verma and A. K. Verma

Abstract—Automatic recognition of machining features is essential for the integration of CAD and CAM. Graph-based recognition is the most researched feature recognition method as the B-Rep CAD modelers' database uses graph to store the model data. A graph-based feature recognition system uses attributed graphs to store CAD models as well as machining feature templates. The graph isomorphism is used to extract features in the model graph and template graphs. There are two main research issues in this system- (1) Efficiently recognize the features as the graph isomorphism is computationally very expensive and (2) incrementally expanding the feature template database to include new features, without any structural change in the recognizer. In this paper, the application of feature vectors (a heuristic developed by the authors that converts a feature graph into a unique vector of integers, irrespective of the node-labeling scheme used by B-Rep modelers), to automatically expand the recognizer's feature template database, is presented. It facilitates automatic inclusion of new features in a feature database, without requiring any additional programming effort from the user or any changes in the structure of the recognizer. The proposed system has been implemented in Visual C++ and ACIS solid modeling toolkit. Further, the proposed system is intelligent as it has the capabilities to learn from the examples to incrementally build the feature database.

Index Terms—Machining feature, feature recognition, graph matching, solid model.

I. INTRODUCTION

An intelligent database is a full-text database that employs some artificial intelligence to return the most relevant information possible requested by the user depending on the context. This is in contrast to a traditional database, which is searchable only by keywords and verbatim phrases with some relational operators. In this paper, definition of intelligent database is extended to include geometrical data along with the textual data. Further a system is presented that interact with this raw geometrical database to extract useful machining features for automated manufacturing of the various parts. The manufacturing process and operation selection is highly dependent of the shape of the part to be produced. Since there are infinite possibilities of shapes it is

not possible to create a predefined library of it, hence shapes are stored in the solid model database in terms of low level geometrical entities such as vertex, edges and faces in the form of a graph or in some other suitable structure. Hence there is a requirement of a system that can extract patterns in this low level database that can be used to automate the manufacturing of the given shape. There are a number of techniques used for this purpose but graph-based recognition is the most researched feature recognition method due to sound mathematical background of graph theory and a graph's structural similarity with B-Rep CAD modelers' database. A graph-based feature recognition system represents geometry and topology information of a part using a graph structure. The graph carries all the useful information of part features as certain attributes assigned to its arcs and/or nodes. This explains the inherent advantage of B-Rep-based solid models over others in graph-based feature recognition. Pre-defined manufacturing features, known as manufacturing primitives are also represented by similar graphs. Graph isomorphism is then used to extract features.

A feature recognition system uses a pre-defined database of feature templates. Thus, a simple mechanism is needed to define new features and expand the template database to include user-defined features without any structural change in the recognizer. This led to development of various methods to represent features. Pratt [1] proposed a non-manifold feature representation scheme using B-Rep for volumetric features. He introduced the concept of implicit and explicit feature representations. In explicit representation, a full geometric shape is defined, while in implicit representation, minimal information is used to define the feature but other details have to be computed when needed. Shah and Rogers [2] and Wang and Ozsoy [3] used hybrid CSG and B-Rep schemes for feature representation. None of these schemes, however, is suitable for feature recognition as the B-Rep or CSG representations of a shape are not unique.

The ASU feature test bed [4], a system developed by Laakko and Mantyla [5], and STEP [International Standard for Exchange of Product data] all use feature definition languages to define new features [6].

In this paper a heuristic has been reported that converts a feature graph into a vector of integers, called feature vector [7], [8]. A feature vector uniquely represents a feature graph, irrespective of the node-labelling scheme used by B-Rep modelers. It can be generated automatically from the modeler database. Its usage is intended to reduce the graph matching time complexity to a polynomial of order three. It also facilitates automatic inclusion of new features in a feature

Manuscript received May 16, 2015; revised November 22, 2015.

Rachna Verma is with the Department of Computer Science and Engineering, J.N.V. University, Jodhpur, India (e-mail: rachna_mbm@yahoo.co.in).

A. K. Verma is with the Department of Production and Industrial Engineering, J.N.V. University, Jodhpur, India (e-mail: arvindrachna@yahoo.com).

library, without requiring any additional programming effort from the user or any changes in the structure of the recognizer.

II. SHAPE AND ITS GRAPH REPRESENTATION

Fig. 1(a) shows a part and Fig. 1(b) represents the part using an attributed adjacency graph (AAG). The solid edges show concave edges and dashed lines show the convex edges. Features in the form of subgraphs of this part useful for manufacturing is shown in Fig. 2 along with their computer representation in the form of attributed adjacency matrix (AAM). The subgraphs are generated by removing all the convex edges and searching for connected graphs in the remaining graph. This part is machinable on a vertical CNC milling machine. Now the task is to extract various features such as holes, slots and packets from this graph and generate relevant instruction for the CNC machine. The CNC program for a feature can be manually generated and stored in the form of a template database. However, a generic part may have different types of features that may not be present in the feature database. For a new pattern that does not exist in the library, the proposed system atomically generate a unique code and store that into the library for future use.

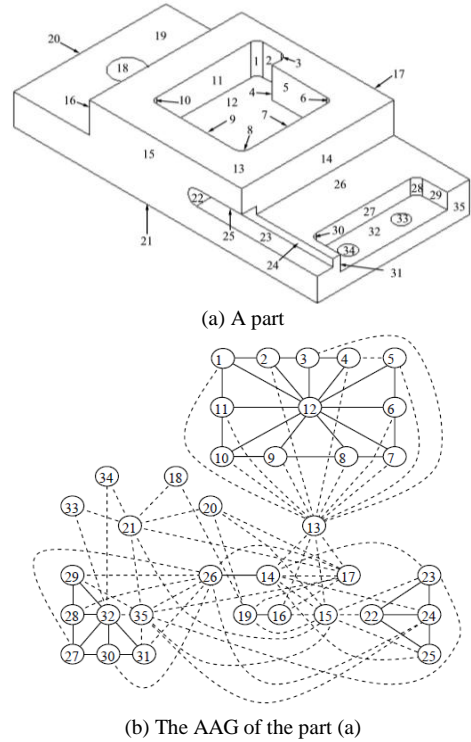


Fig. 1. A part and its attributed adjacency graph.

	Subgraph	AAM					FV	Recognised as								
a			14	26			$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$	Step								
b			16	19			$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$	Step								
c			22	23	24	25	$\begin{bmatrix} 11 \\ 7 \\ 12 \\ 12 \end{bmatrix}$	Blind Slot								
d			27	28	29	30	31	32	$\begin{bmatrix} 31 \\ 56 \\ 38 \\ 37 \\ 48 \\ 40 \end{bmatrix}$	Edge Pocket						
e			1	2	3	4	5	6	7	8	9	10	11	12	$\begin{bmatrix} 2047, 3584, 3088, 2562, 2432, 2308, 2144, 2116, 2072, 2057, 2176, 2080 \end{bmatrix}$	Generic pocket

Fig. 2. Sub graphs, AAMs and feature vectors.

III. FEATURE VECTOR GENERATION HEURISTIC

A heuristic to generate feature vector from an AAM of a

subgraph is described in the following steps.

- 1) Rows and columns of AAM (i.e. vertices of subgraph) are arranged in descending order of their degrees. Degree

of a vertex is the number of proper edges incident with it plus twice the number of self-loops at it. In the present context, there are no self-loops.

- 2) The arranged AAM is partitioned into overlapping zones. Each zone contains those rows and columns that have the same degree. Intersection of rows and columns of a zone forms its pivot block.

- 3) For each zone:

Rows are arranged from top to bottom in descending order of number of 1s present in the pivot block of that zone.

Columns are arranged from left to right in descending order of the column value, treating 1 and 0 as binary numbers, and top row as the most significant bit.

Similarly, rows are re-arranged from top to bottom in a descending order of the row value, taking first column as the most significant bit.

- 4) Finally, considering each row of the AAM as a binary number, feature vector is computed as the ordered set of decimal integer values of the rows.

This arrangement of an AAM uniquely represents an AAG, irrespective of node labelling scheme used in the AAG. To match any two AAGs, their AMMs are converted to respective feature vectors as discussed above. Isomorphism checking of the two AAGs then becomes a matter of comparing the two feature vectors.

subgraphs in an arbitrary part graph. Architecture of the proposed feature recognition system based on feature vector generation heuristic is shown in Fig. 3. It is implemented in Microsoft® Visual C++ along with built-in application programming interfaces (APIs) of ACIS® 3D solid modeling toolkit. The APIs are used for loading, saving, topological and geometric reasoning.

The architecture is built on the following modules.

Module 1. AAG of a part is generated from its B-Rep model and stored as an AAM. Edges of the part are obtained by calling `api_get_edges ()`. Adjacent faces of each edge are found by following the links of B-Rep data structure of ACIS and calling `api_get_faces ()` with edge as its parameter. Convexity of each edge is calculated at its mid point by calling `api_edge_convexity_param ()`. It is assumed that the convexity remains the same throughout the length of the edge, which is true for all milled parts with only analytical surfaces.

Module 2. Subgraphs from the part AAG are generated using standard vertex fusion method.

Module 3. Feature vector corresponding to each subgraph having more than one face is generated. Single node features such as through cylindrical hole can be handled separately.

Module 4. The generated feature vector is matched with templates in the feature library. In the proposed architecture, feature library is stored as a text file and loaded in computer memory at the start of recognition process. It is updated whenever a user defined feature is appended to it. The modified library is reloaded in the memory. Most of the features have unique AAGs; however, there exist some situations where an AAG may represent more than one feature. Specific features in this situation are differentiated by using additional geometric reasoning which can be provided by maintaining link with ACIS database. Presently, such geometric rules are hard coded and require modification of the source code in order to be included in the feature library. A recognized feature is reported as a group of pointers pointing to the faces constituting the feature. This facilitates future analysis of the feature such as manufacturability analysis.

Module 5. If module 4 fails to recognize a cluster of faces, the corresponding feature vector is analyzed to extract generic pocket.

Module 6. Unrecognized cluster of faces not recognized by the previous modules is handled in this module. The user is informed about no-match situation with template features in the library. The system hands over control to the user for breaking the compound feature into primitives. If he feels appropriate, this cluster of faces can be added to the library. In future this feature will be available as a template feature.

IV. ARCHITECTURE OF FEATURE RECOGNIZER AND ITS IMPLEMENTATION

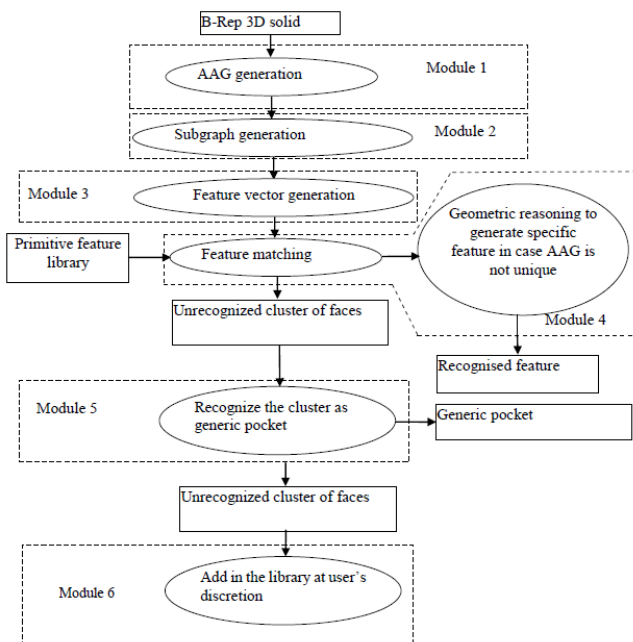


Fig. 3. Architecture of the proposed system.

Features in a part are clusters of faces having some engineering significance. Manufacturing features are subclasses of features that have some manufacturing significance. They are used for planning manufacturing activities. In graph theory terminology, if a part is represented by a graph then its features are subgraphs of the part graph representing pre-defined shapes. The graph-based feature recognition is the process of searching these pre-defined

V. CONCLUSION

Feature vector is a unique representation scheme for a feature AAG. It converts an arbitrarily labeled AAG into a unique vector of integers. It can be automatically generated from a B-Rep model. This helps in incrementally building a feature library as per requirements of specific domain. The proposed system is intelligent enough to extract useful information from the raw database of solid modeler and has

the capabilities to include new patterns found in the data of its own.

REFERENCES

- [1] M. J. Pratt, "A hybrid feature based modelling system," in *Advanced Geometric Modelling for Engineering Applications*, F. L. Krause and H. Jansen, Eds. Elsevier Science Publishers, 1990, ch. 14, pp. 189-201.
- [2] J. J. Shah and M. T. Rogers, "Expert form feature modelling shell," *Computer Aided Design*, vol. 20, no. 9, pp. 515-524, 1988.
- [3] N. Wang and T. M. Ozsoy, "A scheme to represent features, dimensions and tolerances in geometric modeling," *J. of Manufacturing Systems*, vol. 10, no. 3, pp. 233-240, 1991.
- [4] J. J. Shah, D. Anderson, Y. S. Kim, and S. Joshi, "A Discourse on geometric feature recognition from CAD models," *J. of Computing and Information Science in Engineering*, vol. 1, no. 1, pp. 41-51, 2001.
- [5] T. Laakko and M. Mantyla, "Feature modelling by incremental feature recognition," *Computer Aided Design*, vol. 25, no. 8, pp. 479-492, 1993.
- [6] E. S. A. Nasr, A. A. Khan, A. M. Alahmari, and H. M. A. Hussein, "A feature recognition system using geometrical reasoning," in *Proc. International Conference on Manufactuirng of Lightweight Components – Manulight*, 2014, pp. 238-243.
- [7] A. K. Verma and S. Rajotia, "Feature vector: A graph based feature recognition methodology," *Int. J. of Production Research*, vol. 42, no. 16, pp. 3219-3234, 2004.

- [8] A. K. Verma and S. Rajotia, "Review of machining feature recognition methodologies," *Int. J. of CIM*, vol. 23, no. 4, pp. 353-368, 2010.



Rachna Verma received the bachelor of science from University of Delhi in 1996, the master of computer applications from MBM Engineering College, Jodhpur, India in 1999 and the PhD form J.N.V University, Jodhpur, India in 2013.

She is currently working as an assistant professor in the Department of Computer Science and Engineering, J.N.V. University, Jodhpur, India.



A. K. Verma received the bachelor of engineering degree in industrial engineering from IIT Roorkee, India in 1992 and the master of engineering in production and industrial systems engineering in 1994 from the same institute. He received his PhD degree from MBM Engineering College, Jodhpur, India in 2005.

He is currently working as an associate professor in the Department of Production and Industrial Engineering, J.N.V. University, Jodhpur, India.