Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer-Aided Design 38 (2006) 845–856
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

Boundary analysis and geometric completion for recognition of interacting machining features

Keyvan Rahmani*, Behrooz Arezoo

Amirkabir University of Technology, Department of Mechanical Engineering, Hafez Ave, P.O. Box 15875-4413 Tehran, Iran

Received 11 February 2005; accepted 29 April 2006

Abstract

Features are the basic elements which transform CAD data into instructions necessary for automatic generation of manufacturing process plans. In this paper, a hybrid of graph-based and hint-based techniques is proposed to automatically extract interacting features from solid models. The graph-based hints generated by this approach are in geometrical and topological compliance with their corresponding features. They indicate whether the feature is 2.5D, floorless or 3D. To reduce the product model complexity while extracting features, a method to remove fillets existing in the boundary of a 2.5D feature is also proposed. Finally, three geometric completion algorithms, namely, Base-Completion, Profile-Completion and 3D-volume generation algorithms are proposed to generate feature volumes. The base-completion and profile-completion algorithms generate maximal volumes for 2.5D features. The 3D volume generation algorithm extracts 3D portions of the part.

Keywords: Feature recognition; CAPP; Graph decomposition; Geometric completion

1. Introduction

Feature recognition, the automatic interpretation of geometric CAD data in terms of manufacturing features, has not yet been significantly implemented in real industrial activities. The literature on feature recognition is full of algorithms proposed to deal with the problem of feature interactions. Many of these algorithms can be categorized as belonging to one of the following three main groups: graph-based, volumetric decomposition and hint-based approaches [1].

The strength of the graph-based approaches in recognizing isolated features is subordinated to their shortcoming in recognizing interacting features. A group of graph-based methods use Attributed Adjacency Graph (AAG) [2] in conjunction with sub-graph isomorphism algorithms. However, these methods reveal difficulties in the recognition of interacting features, since interactions may cause some of the arcs in the part graph to be missed. There has been some attempts to restore missed arcs using evidential reasoning [3–5], though they were limited to polyhedral objects, without guaranteeing the recovery of the correct set of missed links. Another group of graph-based approaches use B-rep graph of the part interactively to find profiles of connected faces. These profiles are then swept and combined along the cutter axis direction to generate machining features [6,7].

The basic idea of the hint-based approaches is to find traces left by the motion of a milling cutter in the part boundary. These traces are then used to generate a feature volume using geometric completion algorithms [8–10]. Hint-based approaches are more successful in recognizing interacting features than the other existing approaches, but they also have some shortcomings. It is quite possible to find traces which are not promising to find a feature. it is also difficult to find suitable traces for some complex features. Moreover, the existing geometric completion algorithms should be further developed to create more complex pocket volumes.

Volumetric decomposition approaches are mainly divided into two sub-groups: convex-hull decomposition [11–13] and cell-based decomposition [14,15]. These methods have some natural drawbacks: they cannot directly be used to generate machining features. They generate form features which should be converted into machining features. These methods are computationally expensive and cannot always guarantee the generation of the correct set of machining features [1].

Yet, generation of a de facto feature recognition system solely using one of the above mentioned approaches has been

* Corresponding author. Tel.: +98 2166405844; fax: +98 2166419736.
E-mail address: keyvan.rahmani@gmail.com (K. Rahmani).

0010-4485/$ - see front matter
doi:10.1016/j.cad.2006.04.015

elusive. What has become evident is that hybrid approaches, specifically those which are made out of the above three categories, can play an important role to handle the drawbacks of existing feature recognition systems. For example the work done by Gao and Shah [16] was an attempt to make a hybrid graph-based and hint-based approach, assuming that feature hints can also be retrieved from a part graph. In their approach, some sub-graph components, called Minimal Condition Sub-Graphs (MCSG), are generated from the part graph, which are regarded as feature hints. After being further processed using extensive geometric reasoning, MCSGs are completed to a recognizable form by restoring their missed links. In fact, in their approach, the completion stage, a requirement of hint-based approaches, is not volumetric: It is graph components which are completed instead of the feature's geometry. However, it can be observed from pure hint-based methods that the information generated by restoring the missed links can be more effectively extracted by implementation of a powerful volumetric completion algorithm.

This paper is also concerned with making a hybrid of graph-based and hint-based methods to recognize interacting features. Here hints are extracted in a graph form, but the feature is completed geometrically. This method uses an AAG which is decomposed to limit and organize the search space for feature hints. Hints, in sub-graph forms, are extracted from the decomposed graph components. A completed volume is then generated for each feature hint using geometric and topological properties of the part.

2. AAG and curved features

The approaches surveyed in the previous section chiefly deal with 2.5D features. However, recognition of free-form surfaces has also been addressed in the literature [17,18]. In this paper, 3D and free-form features are also considered as a part of the hybrid approach being proposed. Since the approach uses an AAG to extract feature hints, it should be explained which attributes are considered in this graph when curved features are presented.

An example of AAG for a polyhedral part is shown in Fig. 1. In this figure all edges are sharply concave or convex. However, a tool motion may generate many smooth edges in the part boundary. Hence, it is important to also include smooth attributes in AAG. A set of definitions for concave/convex faces, edges, vertices and smooth edges can be found in [19]. However, those definitions are only applicable to 2.5D parts composed of simple quadratic surfaces (planar, cylindrical, spherical and toroidal). For example, a smooth-concave edge is defined as the edge tangentially connecting a concave face to a neutral (planar) or concave face. This is not always true if the part is not 2.5D, as can be seen in Fig. 2, where edge e is not really concave, but convex. A complete definition for the convexity of an edge should take the edge itself into account, not the attribute of its corresponding faces: an edge is smooth-convex when an infinitesimal path moved by a particle on two neighboring faces in a direction perpendicular to the edge is smooth-convex. When used in an AAG, smooth-convex edges

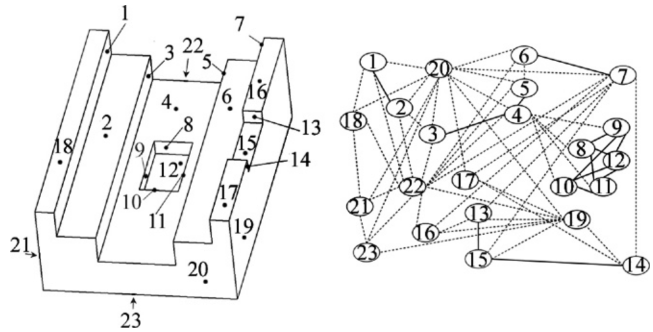

Fig. 1. An example part and its AAG. Solid lines represent concave edges and dashed lines represent convex edges.

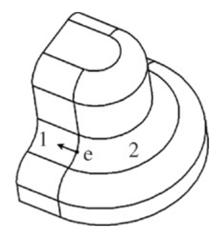

Fig. 2. Smooth edges.

are treated as convex edges and smooth-concave edges treated as concave edges.

In addition, when a part is non-polyhedral, particularly when it includes some free-form surfaces, it is generally possible to have an edge with a changing attribute, although such a change can only be smooth:

Let f and g be two arbitrary faces of a solid model, connected through edge e, having the differentiable underlying surfaces $f(x, y, z) = c$ and $g(x, y, z) = d$. The attribute of the edge at a point $P \in e$ can be calculated using the continues function $h(P) = (\vec{\nabla} f|_P \times \vec{\nabla} g|_P) \cdot \overrightarrow{Co}_f|_P$ where $\vec{\nabla} f|_P$ and $\vec{\nabla} g|_P$ represent the gradient vectors of faces f and g at P, pointing outward from the solid material. $\overrightarrow{Co}_f$ is the co-edge direction vector on face f. By definition, e is convex at P if $h(P) > 0$, and concave if $h(P) < 0$.

If the attribute of e changes, for example from convex to concave, then there is a point $P_0 \in e$ where:

$$\lim_{P \to P0-} h(P) > 0 \quad \text{and} \quad \lim_{P \to P0+} h(P) < 0.$$

However, this is not possible unless $h(P_0) = 0$, which means the edge must be smooth at P_0.

In this research, only the edges with constant attributes are considered. These edges are very common in most practical cases. Edges in 2.5D features have a constant attribute. For free form surfaces, only if one portion of the edge is smooth-concave can it be treated as a concave edge in AAG. When two free-form surfaces join at a smooth concave edge, then the generated geometry of the part at this region can have an effect on selection of a suitable geometry for the cutting tool.

3. Extraction of feature hints

Hints are defined in this research in the form of simple graphs carrying information about a feature's base and side