Feature-based Techniques for Handling Geometric Models

Dinesh Shikhare
National Centre for Software Technology, Juhu, Mumbai.

November 1999.

Abstract

In this survey report we present a study of how features can be used to solve the problems in handling 3D geometric models. We outline the basic aspects of feature based processing of models and discuss some representative application areas. The report also lists the current areas of research in this subject.

1 Introduction

Over the last few years, use of 3D geometric models have been used increasingly in various fields. The dominant fields have been CAD/CAM, virtual reality, digital prototyping and entertainment. The requirement of geometric data processing in all the areas vary due to the different end goals. However, the pursuit to automate processing of geometric data and minimising manual efforts to handle complex and large geometric data has been persistent. This requires automatic adaptation of the geometric algorithms to the specific features in the given models. For example, the research on computer aided process planning (CAPP) [11] has been concentrating on the recognition of features in solid models for the purpose of planning a sequence of machining operations. The machining operations are typically milling, drilling, turning, and so on. Similar examples can be found in literature in the areas of computer-aided design, mesh generation, and other areas.

In this report, we present the study of basic techniques and theory in use for feature based techniques applied to geometric models.

1.1 Some definitions

Features have been defined by various scientists in broad contexts. Some definitions follow:

- “A feature is a region of interest on the surface of a part.” — Pratt, 1985 [9].
- “Features are defined as geometric and topological patterns of interest in a part model and which represent high level entities useful in part analysis.” – Henderson, 1990 [4].

The main difficulty here is that, in trying to be general enough to cover all possibilities, these definitions fail to pin things down sufficiently to give a clear picture. To make these definitions more concrete, we give a classification of features:

- functional feature: for example, a pivot,
- design feature: a rotating pin supported by two raised lugs,
- manufacturing feature: a turned cylinder, a milled slot with an in-line reamed hole through both walls, etc.
- application specific feature: these could be any combination of topological, geometric, metric, colour and texture attributes or non-visual features.

It is important to note that for a given 3D model, different sets of features may be extracted. The features identified for a model in the design feature-space could be much different from those recognised in the manufacturing feature space. For different applications, the model under consideration may be described in different feature spaces.

1.2 Various aspects of use of features

In order to make use of features in processing geometric models, some core common aspects exist. These apply irrespective of the domain of problems and the particular problems at hand. The relevance of individual aspects may vary. These are:

1. *Identification of features:* A particular feature of interest in a given 3D model must be specified in a way such that it can be algorithmically identified in the 3D data. This definition can be a mix of topological properties, geometric metrics, colour and texture attributes and so on. Some higher-level features may be defined in terms of simpler low-level features. Such hierarchies of features are common in literature.
2. *Recognition of features:* Recognition of a feature in the given 3D model is an essential computational part feature-based processing of geometric models. For detecting an individual model, typically specialised filters need to be implemented.
3. *Suppressing features:* By suppressing a feature we mean removing a local instance of the feature while minimally disturbing the data around the feature.
4. *Reconstruction of features:* This is an inverse of “suppressing features.” By reconstruction we mean restoration of a previously suppressed feature in the data, preferably, without any loss of information of the original data.
5. *Encoding of features:* A machine representation of features, preferably a compact one, must be developed.
6. *Feature space conversion:* Given a model described in one feature space, often there is a requirement to convert the description to another feature space.

1.3 Features for what?

As noted above, the specific definitions of features are dependent on the task at hand and also the representation of the 3D data. For example, 3D models designed using CSG techniques will have definitions of features that are completely different from B-rep models constructed using the bounding surfaces.

Here, we cite some examples of applications and the features in use for processing the geometric models.

1. **Features for CAPP:** For manufacture, feature information can be considered to be about volumes of material to be removed or to be added, depending upon the manufacturing process being considered. For material removal processes such as milling, the features can be associated with manufacturing operations and cutters. For example, simple planar slots can be considered as machine operations and tee slots can be considered special cutter operations. These features can be semi-automatically transferred to process planning packages, where upon their cutting paths can be calculated and the possibility of collisions addressed. This allows the machine tools to be driven on the shop floor without the need for skilled process-planners.

   The use of features is not limited to material removal processes. For example, material addition processes such as casting and welding can also have corresponding features.

2. **Features for finite-element analysis:** To carry out finite-element analysis of parts of engineering models, there is a need to discretize the 3D models into simplices. For example, the given surface description of a given model may be transformed into a triangulated surface description. The complexity of the finite-element method increases with the increasing element (triangle) count. The triangle count substantially increases with the increasing surface features on the model. Very often from the viewpoint of the FEM analysis, these features have an insignificant effect. Hence the engineers choose to supress many small features before discretising the models for analysis.

   Automatic simplification of features of geometric models is extremely useful for faster turn-around times in analysis of models [1].

3. **Features for healing of CAD data:** The geometric models created using interactive modelling packages or acquired using 3D scanners often have some artifacts or defects. These defects could be gaps/holes in surfaces, foldover of geometric surfaces, inconsistent orientation of surfaces, coinciding or overlapping triangles, and so on. Looking for and healing such defects manually can be an extremely laborious task. Automatic detection and correction of such defects is needed before using the CAD data for the end applications. Most CAD-data repair packages identify such defects as features to be recognised in given models and to be healed. CADfix is a well known example of a tool developed for such operations [7].

We note that features are often tied to “operators” that can introduce or suppress the geometric/topological characteristics they represent. However, it is also important that the there be an algorithm to mechanically recognize the feature and also a general representation of the feature.

2 Survey of Feature Research

In this section, we survey the representative work of researchers in the area of feature-based processing of geometric models. We begin with a taxonomy of features. These taxonomies do not give us a way finding new features, but once they are found, we classify them and then use the wealth of knowledge in handling the class of features. Later we describe the various approaches taken to solve real problems using features.

2.1 Taxonomy of Features

Many types of classifications have been presented by different researchers [3]. In the CAM community, there has been a rough consensus on distinguishing between *form features* and *machining features*. A form feature, also known as a shape feature, refers to a shape macro constructed for convenience of construction, with little connection with function or manufacturing. A form feature may be either *positive* or *negative*, depending on whether it is a protrusion or a depression, respectively. A machining feature is a negative form feature associated with a distinctive machining process.

A lot of feature research has been concerned with interpreting the CAD data in terms of matchining features and focuses on three machining features: holes, slots and pockets. A hole is generated by a vertical sweep of a drilling cutter. Typically, slots and pockets are made by milling cutters. A slot is usually machined by a single linear sweep of a cyndrical end-milling cutter. A pocket is machined by a series of cuts with an end-milling cutter. We assume (simplistically) that the cutter has a flat end. Then, the pocket feature is represented by an arbitrary shaped planar profile and a sweeping vector.

Dixon et al [2] have classified design features as follows:

- Static Features
    1. Primitive
    2. Intersections
    3. Add-on
    4. Macros
    5. Whole-forms
- Kinetic Features

Pratt [8] presents his classification of form features as:

- Implicit Features
    1. Modifier

        (a) Face
        (b) Edge
        (c) Vertex
    2. Generic
        (a) Prismatic
        (b) Rotational
        (c) Sweep: linear, rotational, other

- Explicit Features
    1. Through hole
        (a) Face: complete or partial
        (b) Edge
        (c) Vertex
    2. Depression
        (a) rotational: complete or partial
        (b) prismatic
    3. Protrusion
        (a) rotational: complete or partial
        (b) prismatic
    4. Area
        (a) with attributes
        (b) without attributes

2.2 Feature Recognition

Feature recognition from solid models has been a subject of research since the 1980s. Either boundary representation (b-rep) or constructive solid geometry (CSG) is typically used for the solid representation of the input part. B-rep based feature recognition has been dominant since, unlike a CSG representation, b-rep uniquely defines the faces of a solid, and so searching for b-rep face patterns is more promising than searching for CSG patterns.

Many approaches have been tried for recognition of features. They include: graph-based matching, convex-hull decomposition, cell-based decomposition and hint-based reasoning.

Graph-based Matching: In this approach [5], the b-rep of a part is translated into a graph whose nodes represent faces and whose arcs represent edges. Additional information may be incorporated into the graph, for example, edge-convexity, face orientation, etc. Primitive features (feature templates) to be searched are also represented by graphs. Subgraph isomorphism is used to search for the subgraphs that match the feature templates.

Subgraph isomorphism is a well-known NP-complete problem. Therefore, the graph pattern matching approach has been criticised for its computational complexity. The criticism may be unwarranted because the template feature graph which to be matched against the part may be very small for matching features like slots, pockets, etc. For these features, n is so small that the asymptotic complexity analysis is not relevant.

The main problem eith the graph pattern matching approach is in its inability to recognize *intersecting features*, not its complexity. The approach has been successful in recognising isolated features. When features intersect, many of the faces of a feature may be entirely absent, partially missing, or fragmoented into several regions.

In addition, the graph pattern matching approach does not ensure the machinability of the recognised features as long as the feature is defined as the collection of faces. The non-volumetric notion of a featuremay cause problems in using this approach for machining applications.

Convex Hull Decomposition: This approach was originally proposed by Woo [13], who used convex hull and set difference operators. The convex hull of CH(P) of a polyhedron P is the smallest convex set containing P. The convex hull difference, or deficiency, CHD(P) is the regularised set difference $(-*)$ between CH(P) and P. Conversely, P can be decomposed as CH(P) $-*$ CHD(P). If P is convex, CHD(P) is empty and the decomposition terminates. Otherwise, the decomposition is applied recursively to the deficiency CHD(P). Woo observed the pattern of alternating volume contributions and called this an Alternating Sum of Volumes (ASV) decomposition. However, it is important to note that ASV decomposition may not necessarily converge. Kim [6] proposed the ASP with Partitioning (ASVP) decomposition for use in generating *form feature* model.

The feature recognition algorithm using the convex hull decomposition is a two-step approach: ASVP decomposition and form feature classification. ASVP decomposition generates a set of volumes (ASVP components) to which feature classification is applied. The ASVP decomposition is completely separated from the feature classification, and is not guided by the goal of recognizing specific types of features.

While this technique can be extended and used to generate machinable instruction for automatic fabrication, it still is not capable of handling intersecting features.

Cell-based Decomposition: This approach was proposed by Sakurai et al [10] and is gaining favour. The delta volume is decomposed into minimal cell by extending and intersecting all the surfaces or halfspaces of he delta volume and then the cells are combined (composed) to generate machining features.

This approach reveals serious problems in both the decomposition and composition steps. The main problem in the cell decomposition is the *global effect of local geometry*. A feature usually leaves traces (faces) in a localized area of the part. However, the cell decomposition step extends the surfaces or halfspaces associated with the faces globally within the delta volume and quite often generates a large number of cells. When we have n cells, all possible combinations constitute its power set. Although some clever heuristics can be applied to prune the set, the algorithm still has exponential complexity.

2.3 Feature-based Design

Here, we briefly overview feature-based design systems. In a Feature-based Design System (FBDS), generic descriptions of features are stored in a feature library. The designer initiates feature instances by specifying dimension/location parameters and various attributes. The features typically are inserted into a workpiece through Boolean operations. FBDSs may be roughly classified into three categories: (a) design with manufactring features, (b) design with form features, and (c) feature modeling combined with solid modeling.

Design with manufacturing features: This approach forces the designer to define a part using a set of features associated with specific manufacturing processes. For machining, the features available to the designer are limited to negative features, all of which are subtracted from the stock.

The advantage of this approach is that the machining features are dirtectly available in the part model and there feature recognition is not needed for planning machining steps. In order to make decisions on process selection, cutter-path generation, etc., process planners can look primarily at local features rather than at the entire path.

However, this approach has some serious drawbacks. Typically, a designer is interested initially only in the *shape and functionality* of the part. The best way to manufacture a part should be determined by downstream applications such as process planning, not by the designer. The "design by manufacturing features" approach assumes that the designerhas ample manufacturing knowledge and forces him/her to transform the design into manufacturing features. This often leads to a drastic shift in thinking process from positive features to negative features.

Design with form features: Ideally, (from a designer's viewpoint) a design should be specified using features that have a true *functional* meaning, but often is specified using positive and negative form features. Many FBDSs using form features have been proposed. However, using the models designed in such system does require a large work of feature recognition to be built so that the design can be streamed down to the machining processes.

Feature modeling with solid modeling: The most flexible design approach is to allow the designer to use whatever is convenient for describing a product. An FBDS may provide a rich library of feature primitives, a powerful ability to modify and combine these primitives, and some capability for user-defined features [2]. However, designers may not want to design a part in terms of features only. A few systems have been proposed where both feature operations and solid modeling operations can be used in parallel during the design of a part [12].

2.4 Feature Model Conversion

Feature model conversion is required across any two different applications that use models described in different feature spaces. The converter must take a model in an application and generate a new feature model for a different application. The application can be design, machining, molding, assembly, etc.

This is an open research area currently in feature-based techniques.

3 Conclusion

From the study carried out in this survey, we conclude that:

1. design and manufacturing features are often different,
2. design should be done in terms of the design features or solid modeling operations, and
3. the part model (which may be a design feature model, or a combination of both) should be converted into a manufacturing feature model.

In feature model conversion, it is now generally accepted that feature recognition is required when direct mapping from design features to manufacturing features is not possible. Hence different aspects of feature recognition continues to be a useful research area to pursue.

References

[1] Jihad El-Sana and Amitabh Varshney. Topology simplification for polygonal virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 4(2), April-June 1998.

[2] J.R. Dixon et. al. Research in designing with features. In H. Yoshikawa and D. Gossard, editors, *Proceedings of IFIP TC5/WG5.2 Workshop on Intelligent CAD*, 1987.

[3] JungHyun Han. Survey of feature research. Technical Report IRIS-96-346, Institute for Robotics and Intelligent Systems, USC, USA, 1996.

[4] M. R. Henderson, S. N. Chuang, P. Ganu, and P. Gavankar. Graph-based feature extraction. Technical report, Arizona State University, 1990.

[5] S. Joshi and T.C. Chang. Graph-based heuristics for recognition of machined features from 3-d solid model. *Computer-Aided Design*, 20:58–66, 1988.

[6] Y. Kim. Recognition of form features using convex decomposition. *Computer-Aided Design*, 24(9):461–476, 1992.

[7] FEGS Ltd. CADFix: A product for CAD data exchange, repair and upgrade. http://www.fegs.co.uk, 1997.

[8] M. J. Pratt. A taxonomy of form features for solid modeling. In M.J. Wozny and H.W. McLaughlin, editors, *Geometric Modeling for CAD Applications*. Elsvier Science, 1988.

[9] M. J. Pratt and P. R. Wilson. Requirements for support of form features in a solid modeling system. Technical Report R-85-ASPP-01, CAM-I, 1985.

[10] H. Sakurai and C. Chin. Definition and recognition of volume features for process planning. In J.J. Shah, M. Mantyla, and D.S. Nau, editors, *Advances in Feature-based Manufacturing*, pages 65–80. Elsvier Science, Amsterdam, The Neatherlands, 1994.

[11] J. J. Shah and M. Mantyla. *Advances in Feature-based CAD/CAM*. John Wiley and Sons, Inc., 1995.

[12] P.C. Sreevalsan and J.J. Shah. Unification of form feature definitio methods. In *IFIP WG5.2 Workshop on Intelligent CAD Systems*, Columbus, Ohio, September 1991.

[13] T. Woo. Feature extraction by volume decomposion. In *Proceedings of CAD/CAM Technology in Mechanical Engineering*, Cambridge, MA, USA, 1982.