

FEATURE MANAGEMENT WITH MULTIPLE VIEWS BASED ON CONVEXITY PROPERTIES OF SHAPE REGION

Relationship and Representation of Design and Manufacturing Features

Hirohisa Kojima

*Department of Aerospace Engineering
Tokyo Metropolitan Institute of Technology
kojima@krasklab6.tmit.ac.jp*

Masatake Higashi

*Department of Mechanical Systems Engineering
Toyota Technological Institute
higashi@toyota-ti.ac.jp*

Abstract This paper proposes a method of representing a product as a feature model with multiple views. A feature is defined as a region which is composed of connected faces by the edges with the same convexity. The method maintains consistent relations between design and manufacturing features against their modification. The model has more than one set of regions and region trees corresponding to an individual view, along with a region graph which represents relations among features in different views. The graph is generated by scanning the common face among the regions in different views. Our method enables detection of the features which can be changed independently from the sequence of the feature creation, and design and manufacturing features on which the shape change has an effect, using the graph and region trees. A feature cancel operation is explained as an example of feature manipulation, which maintains the correspondence of regions using the region Euler operations.

Keywords: Convexity Property, Feature Recognition, Feature Cancellation, Multiple View

1. INTRODUCTION

A feature based modeling system is the major system which supports product development activities such as product design, process planning and product analysis. The model enriches product data representation with semantics information represented by features in some formats. For example, in process planning for manufacturing, features are expressed as volumes that are obtained on machining a blank shape¹⁻⁴. In the design process, parametric modeling, which changes a shape automatically by dimension values, becomes the main method for generating a product model, where features are treated as classes with their dimensions in the slot of the frame representation⁵. However, the present feature-based systems are not suitable for representing the relationship of the features in different engineering views, because correspondence between the volumes and the classes is not explicitly described in the feature model.

In order to deal with the above problems, the Standard for the Exchange of Product Model Data (STEP) gives guidelines on development of product model and exchange of product data between engineering processes. Unfortunately, because of difficulty in defining common features in different engineering activities, its application is still limited in exchanging shape data, and high-level representations including semantics are not shared between design and manufacturing activities.

However, in research to achieve high-quality and the speed-up of the design/production, many studies⁶⁻¹² have been done on integrating features in the design and manufacturing processes. Laako and Mäntylä introduced incremental feature recognition (IFR)⁸, which executes feature recognition after each sequence of geometric modeling operations. It can interpret only the part where a Boolean operation is performed. In their method, features are treated as sub-graphs whose nodes are faces, called attribute adjacency graph(AAG)⁹. Another example is an incremental and localized update of ASVP (Alternation Sum of Volumes with Partitioning) proposed by Parinte and Kim¹⁰. Their method updates the ASVP tree by recognizing a node in the tree which corresponds to a changed part. This is performed by re-decomposing the part, and then re-combining ASVP components. These two studies are efficient for saving time for the model generation and feature recognition by incremental and localized update of the feature model. However, they lack flexibility to represent features with multiple views, and cannot maintain the relation between them. In contrast to the above papers, Inui proposed a method which maintains correspondence between features by checking and correcting the inconsistency between them after feature

modeling operation is done¹¹. However, the method treats only features which can be converted into manufacturing features from design features directly, and does not treat features in other views.

Kraker, on the other hand, proposed a multiple-way feature conversion method which aims at supporting concurrent engineering¹². In his method, a feature model is represented at three levels: feature, canonical shape and evaluated geometry level. The feature level expresses constraint relations between features in different views. The canonical shape corresponds to the node in CSG in one view. At the evaluated geometry level, the model is represented as a cellular model. He does not explain the maintenance method in the feature level, but in the cellular model level. The method, however, has a problem that the number of cells increases when many features intersect.

In order to improve the present feature-based systems, we propose a new method which represents a product model as a feature model with multiple views, by clarifying geometric characteristics which are deduced from the edges with the same convexity. The model has more than one set of features and feature trees along with a feature dependency graph. Here, features are defined as regions which are composed of connected faces by the edges with the same convexity. The feature tree represents a relation among the regions in one view as a tree structure. The tree in the design aspect shows the shape generation procedure and the tree in the manufacturing aspect is used for representing the manufacturing sequence. We can add another view, in addition to the design and manufacturing views. A feature dependency graph, which represents the relations between features in the different views, is obtained by scanning the common face among the regions in different views. The region is expressed by a similar structure to the half edge data structure¹³, and manipulated by a set of operations introduced as region Euler operations.

Our method has following advantage. First, correspondence of the feature model in one view is maintained automatically by region Euler operations, instead of checking the correspondence of regions and their boundaries after the feature has changed. We can apply this idea to the feature model with multiple views. Second, the method enables detection of the features which can be changed independently from the sequence of the feature creation, as well as the design and manufacturing features on which the shape change has an effect, using the graph and trees directly. Both design features and manufacturing features are represented in a product model in the same form as the regions. This detection is an important technology to support the concurrent engineering.

In this paper, we introduce a feature model with multiple views, utilizing regions for each view in section 2. We first explain convexity properties and define a feature region using them. Next, we explain the data structure of the region, its boundaries and region Euler operations, which are basic functions for feature manipulation such as feature recognition and feature cancellation. In section 3, we describe a method of detecting the features which can be changed independently from the sequence of the feature creation, and design and manufacturing features on which shape change has an effect. In section 4, we describe a feature cancel operation as an example of feature manipulation, where a maintenance method for making consistent feature regions is explained. In section 5, we show some examples obtained by our method to demonstrate the functions of the method. Finally, in section 6, we summarize our method and describe future work.

2. REPRESENTATION OF FEATURE MODEL

2.1. DATA STRUCTURE OF FEATURE MODEL WITH MULTIPLE VIEWS

We define a feature as a region which is composed of connected faces by the edges with the same convexity. The region is classified into two types, additive feature region and subtractive feature region. An additive feature has only convex edges between faces, while a subtractive feature has only concave edges between faces. According to the definition, a convex face becomes a top face of a boss feature or face in a base feature, and a concave face becomes a bottom face of a pocket feature (Fig.1). A mixed face which has both convex and concave edges in its outer loop can be interpreted as a part of additive feature or subtractive feature (Fig.2(a)). By utilizing such multiple interpretations for features, we represent a product model as a feature model with multiple views. A product model has more than one set of regions and region trees corre-

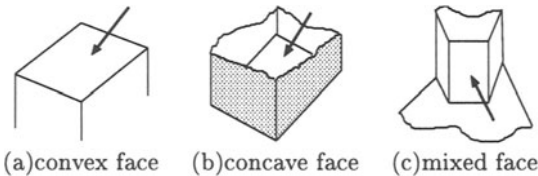


Figure 1 Convexity of faces

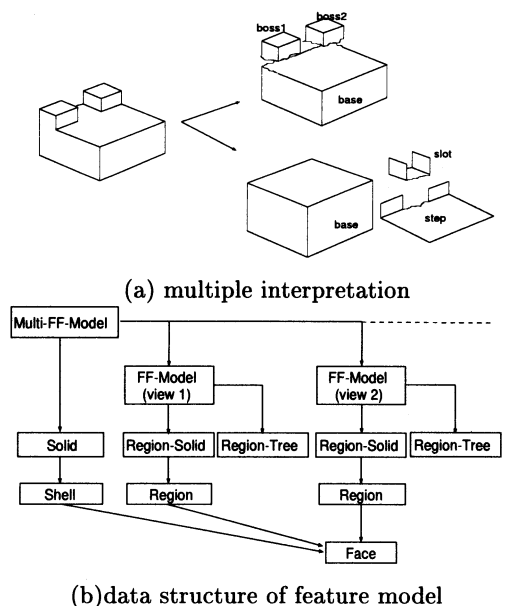


Figure 2 Feature model in multiple views

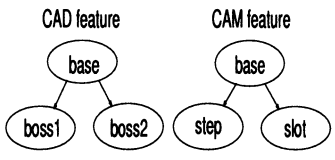
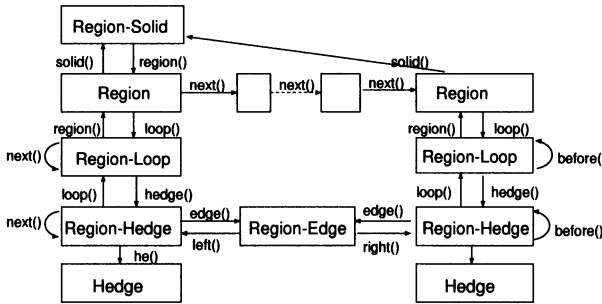


Figure 3 Tree of features in design and manufacturing views

sponding to an individual view, along with a feature dependency graph. In this model, a mixed face belongs to some regions corresponding to an individual view such as design and manufacturing simultaneously (Fig.2(b)). Region trees can be used to represent some sequences, such as a design process and a manufacturing process, and are generated by traversing the mate relation of regions in individual views. In the region tree in a manufacturing view, nodes represent subtractive features except a root, and the root node is a blank shape. The region tree in a design view is regarded as the procedure of shape generation if the reference relation of dimensions is not considered. Fig.2 shows an example of multiple interpretations. It is possible to interpret a given shape generated by adding two boss features to a base feature, or machining

Region data structure		half edge	
Region-Solid	(rs)	Solid	(s)
Region	(r)	Face	(f)
Region-Loop	(rl)	Loop	(l)
Region-Edge	(re)	Edge	(e)
Region-Hedge	(rhe)	Hedge	(he)

(a)Relation between region data structure and half edge data structure



(b)Relation between entities of region data structure

Figure 4 Region data structure

a slot feature and a step feature from the blank shape. Fig. 3 shows multiple feature trees corresponding to the multiple interpretations.

2.2. REGION DATA STRUCTURE

We introduce the following data structure, which is similar to the half edge data structure in order to access and generate regions and their boundaries efficiently. A region data structure is a kind of the boundary data structure. A region, a boundary of a region, a region-edge and a region-halfedge correspond to a face, a face loop, an edge and a halfedge in the half edge data structure (Fig.4(a)). We explain their topology. A region has a set of connected faces and a set of boundaries. A boundary of a region has a pointer to its parent region and a pointer to one of the region-halfedges that forms the boundary. A region-halfedge has a pointer to its parent boundary, a pointer to the region-edge and pointers to the next and previous region-halfedges. A region-edge has pointers to the left and right region-halfedges. A boundary of a region is materialized as a doubly-linked list of region-halfedges. Functions shown in Fig.4(b) are prepared to access the adjacent topological entities.

Region Euler operations	Euler operation
make region r-solid(new)	make vertex face shell
kill region r-solid(delete)	kill vertex face shell
split r-edge	split edge make vertex
join r-edge	join edge kill vertex
make r-edge region	make edge face
make r-loop region	
kill r-edge region	kill edge face
kill r-loop region	
make r-edge kill r-loop	make edge kill ring
kill r-edge make r-loop	kill edge make ring
make region kill r-loop hole	make face kill ring hole
kill region make r-loop hole	kill face make ring hole

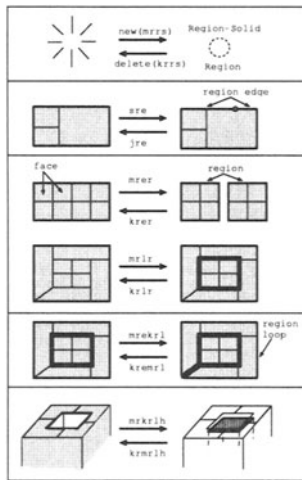


Figure 5 Region Euler operations

2.3. REGION EULER OPERATIONS

We prepared twelve operations to manipulate a region, which are similar to the Euler operations to face. We called them *region Euler operations*. Fig.5 shows the relation between region Euler operation to the region data structure and Euler operations to the half edge data structure. For example, “make r-edge region” corresponds to “make edge face”. Consistency of the region, such as the number of faces in the regions and their boundaries, is automatically maintained by the

operations, because (region) Euler operations cannot create an invalid boundary data structure. These operations are used in decomposing a model into feature regions¹⁴ and in a procedure of feature cancel operations. The feature cancel operation will be explained in section 4.

3. RELATION BETWEEN DESIGN AND MANUFACTURING FEATURES

3.1. FEATURE DEPENDENCY GRAPH IN DIFFERENT VIEWS

Design features are regions which are generated in the design process. Manufacturing features are regions which are recognized only as subtractive types. The relation of design and manufacturing features are obtained by scanning the common faces in the design and manufacturing features. For the example shown in Fig.6, when faces f_1 and f_2 , belonging to design feature A, existing in manufacturing feature B, C, we generate a graph whose node A (corresponding to design feature A) has arcs connected to node B, C (corresponding to manufacturing feature B, C). We call the graph “feature dependency graph” (Fig.6). The graph is utilized for detecting the features on which shape change has an effect.

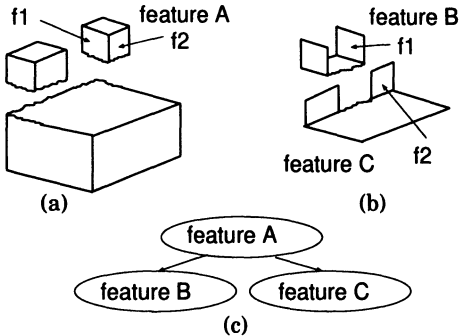


Figure 6 Dependency of features in multiple views

3.2. DETECTION OF A PART WHICH CAN BE CHANGED INDEPENDENTLY FROM DESIGN SEQUENCES

In order to achieve a shape change efficiently, we introduce a method which detects the feature that can be changed independently from the sequences of feature creation. Since the region trees are regarded as a sequence of shape generation, the end node in the tree represents a part which is not interfered by other features. Therefore, features which are the end nodes both in design and manufacturing trees can be canceled or generated independently from the sequence of feature creation. These parts can be detected by retrieving the end nodes in the trees.

3.3. DETECTION OF RANGE BE AFFECTED BY SHAPE CHANGES

To inspect whether the design change is appropriate to the design purpose and productivity, we must estimate other features which the change affects. This can be done by using the feature dependency graph and the feature trees. The features affected are classified into three types, as follows:

- (a) Design feature which is designated for change,
- (b) Manufacturing features whose faces belong to the feature (a), (We can discover these features by scanning the feature dependency graph.)
- (c) Design and manufacturing features which have an indirect relation with the features (b). (We can find these features by checking the parent-child relationship in the trees.)

Fig.7 shows an example of detection of features on which the change of boss 1 feature has an effect. As shown in Fig.7, a base feature, a slot feature and a step feature in manufacturing view are features which are influenced directly, and the base feature and boss2 feature in the design view are features which are influenced indirectly.

4. MAINTENANCE OF FEATURE REGIONS IN FEATURE CANCEL OPERATION

In the process of parametric feature modeling and generation of intermediate shapes for the manufacturing process, features may be re-generated or canceled from a shape. In these processes, it is important

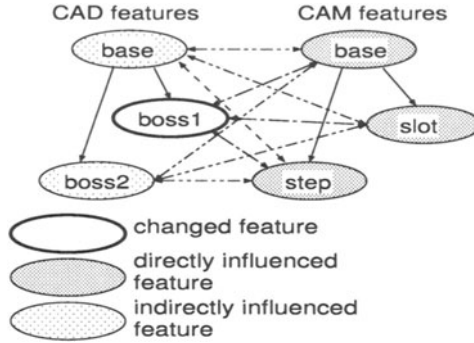


Figure 7 Detection of influenced features

to maintain the consistency of feature regions in order to preserve relation between features in different views. We explain the maintenance method in the case of feature cancel operations. The feature cancel operation, which removes a feature from a shape, is executed step by step from descendants to parents, according to the region tree. This result can be used to process planning directly because it represents steps of processes and the intermediate shapes in them. The method detects how the boundaries of the region change every time one feature is canceled, and then maintains the faces and boundaries of the regions adjacent to the canceled region. The feature cancel operation is performed as follows.

- 1 We first generate a lamina shell by extending the surrounding faces adjacent to a feature which will be canceled⁴. The extension is executed to hold the connect condition between faces in the feature region. The generated shell might become a concave shape (Fig.8(b)), while the traditional method which uses half spaces only generates a convex shell.
- 2 A lamina shell consists of three types of regions (Fig.9). A back side of the shell which becomes faces of feature volume (Fig.9(a)), an interference region which is generated by the feature interference (Fig.9(c)), and the other region which is a part of the front side in the shell without the interference region(Fig.9(b)). We recognize feature interference parts in the shell by using the face convexity properties, and then, we decompose the front side of the shell into two regions using the “make r-edge region” operation.

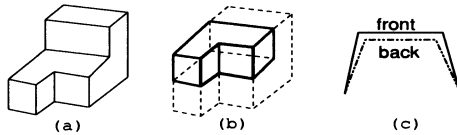


Figure 8 Given object(a) and lamina shell (b)(c)

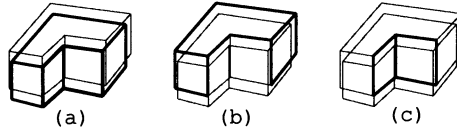


Figure 9 Classification of regions in lamina shell

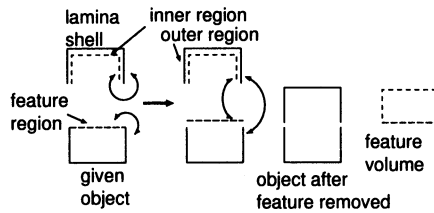


Figure 10 Sewing two shells along region boundary

This step is equal to the step in decomposing a shape into regions for feature recognition¹⁴.

- 3 We exchange connections between the shell and the given shape (Fig.10). In the case where boundaries between regions, which are merged, are adjacent to a different region, we execute the “krer (kill r-edge region)” operation. If this is not the case, we perform the “kremrl (kill r-edge make r-loop)” operation. This step corresponds to the reverse step of the decomposition in the feature recognition. At this step, a feature volume is generated.
- 4 Lastly, we sew faces in the front side region and the faces outside of the canceled feature region along the boundaries of regions using

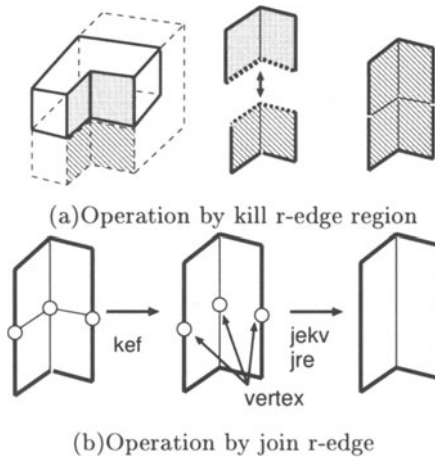


Figure 11 Region connect operations

the “kef (kill edge face)” operation. And then, we connect the edges using the “jekv (joint edge kill vertex)” and “jr-e(joint r-edge)” operations(Fig.11).

Step 3 and step 4 are similar to the topological processes in the Boolean operation for solid modeling. In order to maintain all features in multiple views, we only execute the above steps to regions in each view, because the number of faces included in the feature and boundaries of regions are automatically maintained.

However, we must maintain the consistency of region trees and the dependency graph, since the mate relations of regions change after the cancel operation is done. The maintenance is executed by traversing the mate relation of regions, as well as scanning the common faces in the different views. This process can be localized in some features which are adjacent to the canceled feature.

5. EXAMPLES

We have implemented the algorithms with GNU C++ language, which is a kind of object-oriented language, on Indigo2 (Silicon Graphics Workstation) for the pointer manipulations of the B-rep model. Namely, topological entities in the region data structure and functions to access them are implemented as classes and methods in the object-oriented language.

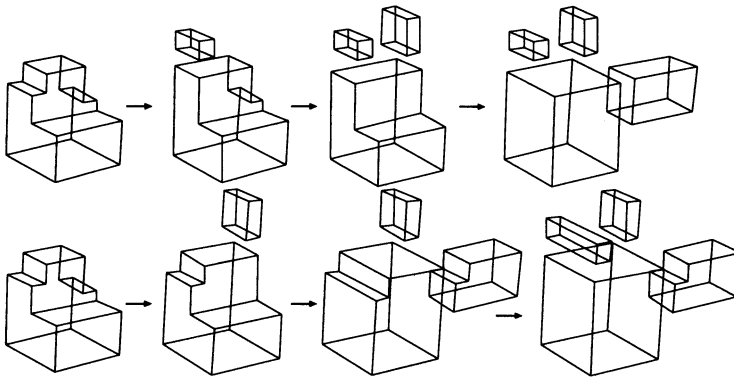


Figure 12 Example of feature cancel operations

5.1. EXAMPLE OF FEATURE CANCEL OPERATIONS

Fig.12 shows multiple cases of feature cancel operations. The shape includes three open-step features. There is freedom for the cancel operation. The number of procedures is six according to all the permutation. Here, two of them are shown as feature volumes generated. Which procedure is selected depends on the application. For example, the upper case would be chosen for the machining, because it selects features according to machinability. Our feature cancel operation method can create concave feature volumes as shown in Fig.12.

5.2. EXAMPLE OF DETECTING AFFECTED RANGE AND MAINTAINING FEATURE REGION TREES

Fig.13 shows an example of detecting the affected range when a slot feature is removed. In this example, three faces, f10, f11, f12 in the slot feature are detected as changing parts. Each face is included in features rib1 and rib2, the method detects rib1 and rib2 as features in the design view which are directly affected. Each range is enclosed by dotted lines and chained lines. In addition, the features which have indirect contact with these features are detected. The features enclosed by solid lines are features which might be affected by the detected features, because they are children of the above-detected features.

Fig.14 shows an example of a region-tree change when a slot feature is cancelled. Two rib features are merged into one rib feature, and two

circular slot features change into the holes in the design view. Two step features and two holes become child regions of the base region in the manufacturing view.

Since our method maintains the consistency of regions and their boundaries, it is not necessary to decompose intermediate shapes into feature regions at each step. However, since feature types change in the cancel operation, it is necessary to re-recognize the type of features whose faces are adjacent to the removed feature.

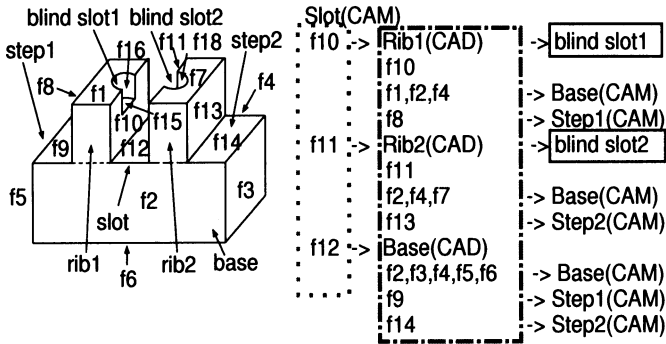


Figure 13 Example of detecting affected range

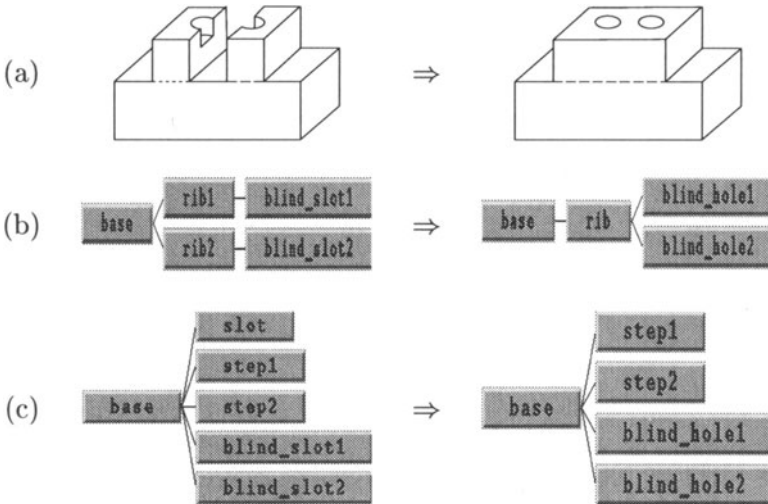


Figure 14 Example of region tree change: object (a), region trees of CAD (b) and CAM (c)

6. SUMMARY

We have proposed a method which represents a product model as a feature model with multiple views. The feature is defined as a region which is composed of connected faces by the edges with the same convexity. The model has more than one set of features, feature trees corresponding to an individual view and a feature dependency graph which represents the relation among features in different views. The region is expressed by a similar structure to the half edge data structure, and manipulated by region Euler operations. A method of maintaining the correspondence of the region in the feature cancellation using region Euler operations, is described. By this method, it becomes possible to cancel features continuously in multiple and different sequences from those of the feature creation. From the representation of the feature model, it becomes possible to express the relation of design features and manufacturing features in the product model, and to detect the features which can be changed independently from the sequence of the feature creation, and the design and manufacturing features on which the shape change has an affect.

Our future research will study application of the method to process planning for machining and assembling, a method of maintaining consistency of feature dimensions in cancellation operations and a method of checking the validity of a machining sequence.

References

- ¹Kim, Y.S. Recognition of form features using convex decompositions. *Computer-Aided Design*, **24**, 9, (1992), pp461-476.
- ²Sakurai, H. Volume decomposition and feature recognition: Part 1—polyhedral objects, *Comuter-Aided Design*, **27**, 11, (1995), pp833-843.
- ³Woo, T. Feature extraction by volume decomposition, *Proc. Conference on CAD/CAM Technology in Mechanical Engineering*, MIT, Cambridge, MA, (1982), pp76-94.
- ⁴Dong, X. and Wozny, M.J. "Feature volume creation for computer aided process planning." In *Geometric Modeling for Product Engineering*, M. J. Wozny, U. U. Turner and L. Preiss(Eds), Elsevier Science Publishers B.V., (1990), pp385-403.
- ⁵Shah, J.J. and Mäntylä, M. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. John Wiley & Sons (1995).
- ⁶Gadh, R. and Prinz, F.B. Recognition of geometric forms using the differntial depth filter. *Comuter-Aided Design*, **24**, 11,(1992), pp583-598.

- ⁷Shah J.J. Feature transformations between application specific feature spaces. *Computer - Aided Engineering Journal*, **5**, 6, (1988), pp247-255.
- ⁸Laako, T. and Mäntylä, M. Feature modeling by incremental feature recognition. *Computer-Aided Design*, **25**, 8, (1993), pp479-492.
- ⁹Joshi, S. and Chang, T.-C. Graph-based heuristics for recognition of machined features from a 3-D solid model. *Computer-Aided Design*, **20**, 2, (1988), pp58-66.
- ¹⁰Parinte, F and Kim, Y. S. Incremental and localized update of convex decomposition used for form feature recognition. *Computer-Aided Design*, **28**, 8, (1996), pp589-602.
- ¹¹Inui, M., Suzuki, H., Kimura, F. and Sata, T. Process Planning Automation Based on the Dynamic Manipulation of Form Feature. *JSPE* **54**, 10, (1988), pp.171-176. in japanese.
- ¹²Klass Jan de Kraker, Maurice Dohmen, M., and Bronsvort, W. F. Maintaing multiple views in feature modeling. *Proc. ACM Solid Modeling*, ACM Press, New York, (1997), pp123-130.
- ¹³Mäntylä, M. *An Introduction to Solid Modeling*. Computer Science Press, Maryland, (1988).
- ¹⁴Kojima, H., Inoue, N. and Higashi, M. "Feature Extraction by Convexity Information in Face-Based Representation." In *Rapid Product Development*. N.Ikawa, T.Kishinami and F.Kimura(Eds), Chapman & Hall, (1997), pp425-434.