
Feature modeling using a grammar representation approach

E. OSTROSI AND M. FERNEY

Laboratoire de Recherche Mécatronique 3M, Université de Technologie de Belfort-Montbéliard, Belfort, France

(RECEIVED October 12, 2004; ACCEPTED May 19, 2005)

Abstract

In intelligent computer-aided design the concept of intelligence is related to that of integration. Using feature-based computer-aided design models is thought to make a complete integration. This paper presents a feature recognition approach based on the use of a feature grammar. Given the complexity of feature recognition in interactions, the basic idea of the approach is to find the latent and logical structure of features in interaction. The approach includes five main phases. The first phase, called regioning, identifies the potential zones for the birth of features. The second phase, called virtual extension, builds links and virtual faces. The third phase, called structuring, transforms the region into a structure compatible with the structure of the features represented by the feature grammar. The fourth phase, called Identification, identifies the features in these zones. The fifth phase, called modeling, represents the model by features. The feature modeling system software is developed based on this approach.

Keywords: Computer-Aided Design; Design; Feature Modeling; Grammar; Recognition

1. INTRODUCTION

Today, computer-aided design (CAD) is used at all levels in the design process, from conceptualization to documentation. The introduction of solid modelers made it possible to unambiguously define the geometric models of parts. This type of modeler provides a complete representation of the shape of the part, offering sufficient information about its geometry and topology. Thus, this type of representation is increasingly popular in various applications used in the product development process.

Despite these advantages, these modelers only take into account the product shape, but do not include the various types of knowledge required to build and utilize it. The introduction of the concept of features made it possible to associate shape and knowledge in understanding a CAD model.

Features are generic or specific shapes with which engineers associate certain attributes and business knowledge used in various development phases. To use features as a

means for integration, we must find ways to transform their representation when moving between applications. This problem involves, on the one hand, transforming a geometric model for the part into a feature-based model adapted to the desired view, and on the other hand, enabling the transformation of features between views (Bronsvort & Jansen, 1993; Laako & Mäntylä, 1993; De Martino & Gianini, 1994). The transformation, which is developed based on a geometric model to other models representing different application views, is called automatic feature recognition.

Automatic feature recognition is a process for transforming one representation into another. More specifically, we can say that the process defines the transformation of all the lower level CAD model entities such as primitives, faces, edges, and nodes into features. Automatic feature recognition must resolve the following problems: feature representation and feature recognition. The first problem involves choosing or developing a approach suitable for representing features so that their representation is unique. The second problem involves developing inference procedures able to perform the most complete recognition possible. Directed to these problems, two classes of feature recognition approaches can be distinguished: decision-theoretical approaches and structure-handling approaches.

Reprint requests to: E. Ostrosi, Laboratoire de Recherche Mécatronique 3M, Université de Technologie de Belfort-Montbéliard, 90010 Belfort Cedex, France. E-mail: egon.ostrosi@utbm.fr

In the decision–theoretical approaches, a feature is represented as a vector, giving information about one face of the feature and its relationships to other faces. A face definition consists of face geometry and a set of bounding edges and vertices. If a value is assigned to the edges and vertices based on their geometric and topological information, then a face score will summarize the important properties of the face. If the face scores are evaluated for all faces in the object, then the collection of face scores describes the local behavior and can form a face score vector. Thus, features can be defined as the region comprising the faces of the object, characterized by a specific face score gradient. The definition of a feature depends on identification of the primary face of a feature, which is defined as the face with the highest score in the defined region (Henderson, 1994). The disadvantage of this approach is its lack of suitable formalism for handling feature structures and their relationships.

In the structure–handling approaches, features are represented based on their logical structures and their relationships. Structure–handling approaches deal with the explicit knowledge, a capability lacked by decision–theoretical approaches. The basic idea of these approaches is to search the logical latent structure of features. Here, the main automatic feature recognition approaches are based on *graph theory* (Ansaldi et al., 1985; Joshi & Chang, 1988; Marefat & Kashyap, 1990), *expert systems* (Henderson & Anderson, 1984; Bond & Chang, 1988; Choi et al., 1988); *volume-based decomposition* (Woo, 1982; Kim, 1992; Kim & Wilde, 1992; Menon & Kim, 1994), and the *syntactic approach* (Srinivasan et al., 1985; Li, 1988; Falcidieno & Gianini, 1989).

Graph theory is a popular method used to represent the topological relationship of features. For instance, the attributed adjacency graph is used for feature representation (Joshi & Chang, 1988). An attributed adjacency graph is defined as $G(N, A, T)$, where N is the set of nodes, A is the set of arcs, and T is the set of attributes assigned to arcs in A . Each face of features is represented as a node, and each edge or face adjacency is shown as an arc. The attribute is assigned “1” if the two adjacent faces form a convex angle and “0” if the angle is concave. For each feature type a rule, usually the first-order one, based on the graph theory is developed.

The expert system approach represents the features by some rules of logic (Henderson & Anderson, 1984; Bond & Chang, 1988; Choi et al., 1988). They are the production rules in the logic of the first order using geometrical and topological relationships (Henderson & Anderson, 1984).

The volume decomposition approach is based on the assumption that a feature can be represented as a volume or a sum of smaller volumes. A convex decomposition called alternating sum of volumes with partitioning is developed (Woo, 1982; Kim, 1992; Kim & Wilde, 1992; Menon & Kim, 1994). There are two stages in that approach: the decomposition, and converting of the decomposition into form features.

The syntactic approach is based on the decomposition of features into subpatterns or primitives. By tracking the fea-

tures patterns in one direction, it is possible to detect and encode these primitives in the form of string of qualifiers. Suppose that one can interpret each primitive as being a symbol permissible in some grammar, where a grammar is a set of rules of syntax for the generation of sentences from the given symbols; then, a language is generated by this grammar. The sentences of this language represent the features (Srinivasan et al., 1985; Li, 1988; Falcidieno & Gianini, 1989).

Automatic feature recognition is a complex process. Despite major developments in this area, several problems remain. Thus, feature recognition in interaction remains an area for research. In the case of design, a finite set of canonical features can produce an unlimited number of configurations of features in physical interaction. The physical interaction between canonical features can deform their initial representation. Indeed, the representation of a feature, resulting from the physical interaction between canonical features, is dissimilar from the representation of canonical features. Furthermore, the representation of a canonical feature as a component of a resulting feature may be different from its initial representation. Listing all the features in interaction appears to be a utopian task of very little interest (Marefat & Kashyap, 1990).

This paper discusses the problem of recognizing canonical features and features in interactions. In the second section, using the hypothesis that a product has a final structure that is the result of an ideal evolution from a set of significant structures, we propose a feature grammar for their representation. Consideration of the semantic and uncertain aspects generalized the feature grammar by producing the conditional and fuzzy features grammar. The third section presents a recognition approach based on the conditional and fuzzy features grammar. Examples and the application illustrate the steps involved and the advantages of this approach.

2. FEATURE REPRESENTATION

2.1. Topological and geometric entity graph

A *feature* is a geometric entity defined by its shape and technological characteristics, typically represented by a set of topologically associated faces. Given two finite, non-empty sets $D^{\text{tp1}} = \{D_1^{\text{tp1}}, D_2^{\text{tp1}} \dots D_m^{\text{tp1}}\}$ and $D^{\text{geo}} = \{D_1^{\text{geo}}, D_2^{\text{geo}} \dots D_n^{\text{geo}}\}$, which are called the set of topological domains and the set of geometric domains, respectively; two sets of attributes $A^{\text{tp1}} = \{a_1^{\text{tp1}}, a_2^{\text{tp1}} \dots a_m^{\text{tp1}}\}$ and $A^{\text{geo}} = \{a_1^{\text{geo}}, a_2^{\text{geo}} \dots a_n^{\text{geo}}\}$, which are called the set of topological attributes and the set of geometric attributes, respectively, where each attribute is associated with each domain; and $X = \{X_1, X_2 \dots X_i \dots X_m\}$, which is a set of features. Then, any shape feature X_i can be characterized by a set of faces $F = \{f_1, f_2 \dots f_i \dots f_m\}$ that satisfy a set of topological and geometric relations. These relations are

Table 1. Domains and associated attributes

| | Topology | | Geometry | | |
|---|--------------------------------------|------------------------------|--------------------------|-----------------------------------|------------------------------|
| | Relative Positions (a_1^{pl}) | Face Type (a_2^{geo}) | Angle (a_1^{geo}) | Adjacency Type (a_2^{geo}) | Face Type (a_3^{geo}) |
| | Domains | | | | |
| 0 | Adjacent, | Base | Convex | Line | Plane |
| 1 | Nonadjacent | Side | Concave | Nonstraight line | Nonplane |
| 2 | Parallel | Frontal | Flat | Other | |
| 3 | Virtual adjacent | | Other | | |
| 4 | Same prolongable support | | | | |
| 5 | Same nonprolongable support | | | | |

defined for domains corresponding to the set of topological and geometric attributes, respectively. Table 1 shows typical cases of those attributes and their respective domains. These relations may be represented by the topological and geometric entity graph, defined as follows:

DEFINITION. For two given sets

$$F^* = \{(f_i, e_i^*) | f_i \in F\},$$

where $F = \{f_1, f_2 \dots f_i \dots f_m\}$ is a set of faces and $e_i^* = (a_2^{pl}, a_3^{geo})$ is an entity associated with each face f_i , and

$$E = \{(f_i, f_j, e_{ij}) | f_i, f_j \in F\},$$

where $e_{ij} = (a_1^{pl}, a_1^{geo}, a_2^{geo})$, we call $G = (F^*, E)$ the topological and geometric entity graph. ■

In the graphical representation of the topological and geometric entity graph, the nodes associated with the label $e_i^* = (a_2^{pl}, a_3^{geo}, a_4^{geo})$ represent the topological and geometric relation of the faces, and the edges associated with the label $e_{ij} = (a_1^{pl}, a_1^{geo}, a_2^{geo})$ represent the topological and geometric relation between a pair of faces (f_i, f_j).

Figure 1 shows the representation of the “Slot” feature by the topological and geometric entity graph.

2.2. Feature grammar

A *feature language* describes the generation of feature structures, joint elements, and attaching elements. A *grammar* provides the finite generic description of this language. Thus, we will focus on finding a *feature grammar*, which provides the generic and productive description of the *feature language*. In these conditions, a feature grammar can be defined as an 8-plet:

$$G_{\text{Feature}} = \{V_{\text{structure}}^t, V_{\text{joint-tie}}^t, V_{\text{structure}}^N, V_{\text{joint-tie}}^N, S, \nabla, \Lambda, P\}, \tag{1}$$

where $V_{\text{structure}}^t = \{a, b, c \dots\}$ is the terminal vocabulary of structures, a nonempty finite set; $V_{\text{joint-tie}}^t = \{0, 1, 2 \dots j \dots m\}$, $m \in N$, is the terminal vocabulary of joint-tie elements, a nonempty finite set; $V_{\text{structure}}^N = \{A, B \dots S \dots\}$ is the nonterminal vocabulary of structures, a nonempty finite set; $V_{\text{joint-tie}}^N = \{O, I, II, III \dots \nabla, \Lambda\}$ is the nonterminal vocabulary of joint-tie elements, a nonempty finite

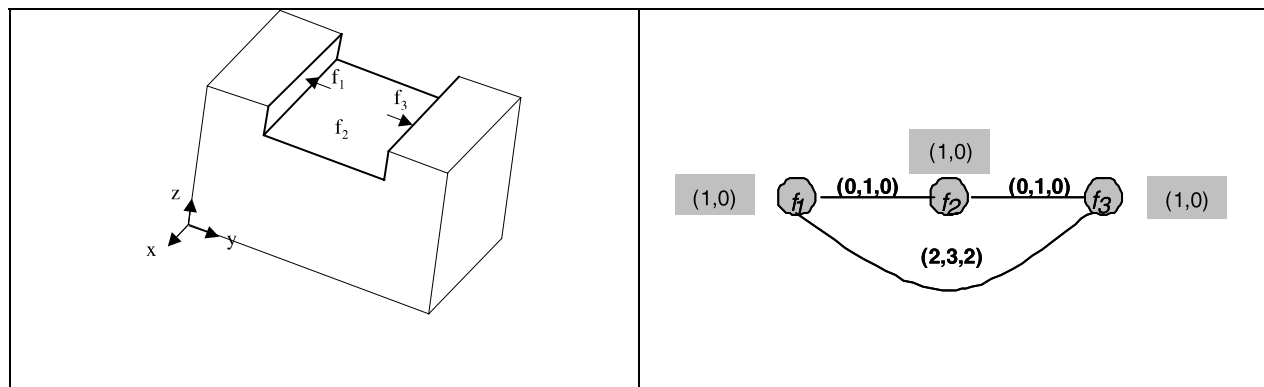


Fig. 1. <Slot> and its topologic and geometric entity graph.

set; \mathcal{S} , ∇ , and Λ are the respective structure, joint, and connection axioms;

$$P: \left\{ \left[\begin{array}{c} \alpha \\ \Gamma_\alpha \\ \Delta_\alpha \end{array} \right] \rightarrow \left[\begin{array}{c} \beta \\ \Gamma_\beta \\ \Delta_\beta \end{array} \right] \right\}$$

is a set of production rules; and

$$V_{\text{structure}}^T \cap V_{\text{structure}}^N = \emptyset,$$

$$V_{\text{joint-tie}}^T \cap V_{\text{joint-tie}}^N = \emptyset,$$

$$(V_{\text{structure}}^T \cup V_{\text{structure}}^N) \cap (V_{\text{joint-tie}}^T \cup V_{\text{joint-tie}}^N) = \emptyset.$$

The production rules of the feature grammar have the following format:

$$\left[\begin{array}{c} \alpha \\ \Gamma_\alpha \\ \Delta_\alpha \end{array} \right] \rightarrow \left[\begin{array}{c} \beta \\ \Gamma_\beta \\ \Delta_\beta \end{array} \right] \quad \begin{array}{l} \text{level 1} \\ \text{level 2} \\ \text{level 3} \end{array} \quad (2)$$

where α is called the left-side component matrix, $\alpha = [\alpha_{ij}]$, $i = 1, j = 1$; β is called the right-side component matrix, $\beta = [\beta_{ij}]$, $i = 1, j = 1, 2 \dots m$; m is the number of components; Γ_α is called the left-side joint matrix, $\Gamma_\alpha = [\Gamma_{\alpha_{ij}}]$, $i = 1, 2 \dots n$; $j = 1$; n is the number of attaching elements; Δ_β is called the right-side joint matrix; $\Gamma_\beta = [\Gamma_{\beta_{ij}}]$, $i = 1, 2 \dots n$; $j = 1, 2 \dots m$; Δ_α is called the left-side tie-point matrix, $\Delta_\alpha = [\Delta_{\alpha_{ij}}]$, $i = 1, 2 \dots s$; $j = 1$; s is the number of attaching elements; and Δ_β is called the right-side tie-point matrix, $\Delta_\beta = [\Delta_{\beta_{ij}}]$, $i = 1, 2 \dots s$; $j = 1, 2 \dots n$.

There are three levels of production rules for the feature grammar. The first is the component level. These rules have the following formats:

$$[\alpha_{ij}] \rightarrow [\beta_{ij}] \quad (3)$$

or

$$[A] \rightarrow [v_1 \ v_2 \ \dots \ v_j \ \dots \ v_m] \quad (4)$$

where $\alpha \in V_{\text{structure}}^N$; $\beta = v_1, v_2 \dots v_j \dots v_m$; this matrix defines an order relation for these components; $v_j \in V_{\text{structure}}^T \cup V_{\text{structure}}^N$ is a terminal or nonterminal structure called the *component structure*; and m is the number of structures.

The second level is the joint level. These rules have the following formats:

$$[\Gamma_{\alpha_{ij}}] \rightarrow [\Gamma_{\beta_{ij}}] \quad (5)$$

or

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} \rightarrow \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1j} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2j} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ t_{i1} & t_{i2} & \dots & t_{ij} & \dots & t_{im} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nj} & \dots & t_{nm} \end{bmatrix}, \quad (6)$$

where y_i is a joint element and t_{ij} is an attaching element of component j , defined according to the order in the right-side component matrix, that participates in forming the y_i .

The third level is the connection level. These rules have the following formats:

$$[\Delta_{\alpha_{ij}}] \rightarrow [\Delta_{\beta_{ij}}]. \quad (7)$$

or

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_s \end{bmatrix} \rightarrow \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1j} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2j} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ t_{i1} & t_{i2} & \dots & t_{ij} & \dots & t_{im} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ t_{s1} & t_{s2} & \dots & t_{sj} & \dots & t_{sm} \end{bmatrix}, \quad (8)$$

where z_i is an attaching element and t_{ij} is an attaching element of component j , defined according to the order in the right-side component matrix, that participates in forming the z_i .

2.2.1. Conditional feature grammar

The feature grammar represents the purely *syntactic* side. It does not always allow the expression of the full complexity of structural relations between the primitive elements composing a feature. If a syntax rule must meet mandatory conditions before being applied, then a conditional feature grammar is defined as follows:

$$G_{\text{Feature}}^C = \{G_{\text{Feature}}, A^{\text{geo-tp1}}, D^{\text{geo-tp1}}, C\}, \quad (9)$$

where G_{feature} is the feature grammar,

$$C = \begin{bmatrix} C_{\alpha \rightarrow \beta} \\ C_{\Gamma_\alpha \rightarrow \Gamma_\beta} \\ C_{\Delta_\alpha \rightarrow \Delta_\beta} \end{bmatrix}$$

are the three levels of semantic conditions, $A_{\text{geo-tp1}}$ is the set of geometric and topological attributes, and $D_{\text{geo-tp1}}$ is the set of geometric and topological domains.

2.2.2. Fuzzy feature grammar

The elements of $V_{\text{structure}}^T = \{a, b, c \dots\}$ that have a certain property, such as the *nonexistence of virtual edges in a terminal a*, make up a subset of $V_{\text{structure}}^T$. If some elements of $V_{\text{structure}}^T$ do not have this property in an absolute

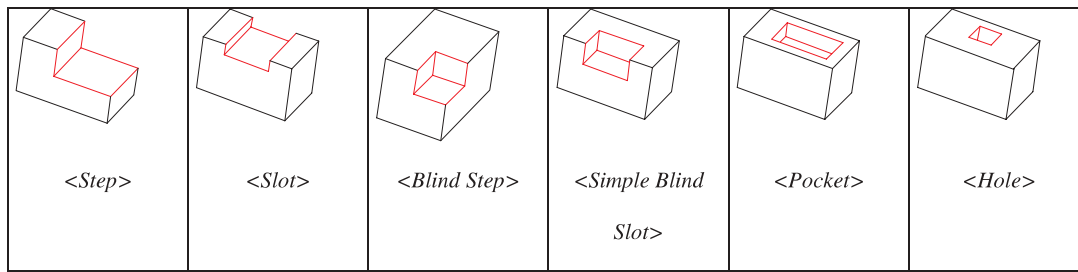


Fig. 2. A sample of the features in the set X. [A color version of this figure can be viewed online at www.journals.cambridge.org]

manner, we may choose to indicate the extent to which each element has the property. Thus, we define a fuzzy subset of $V_{structure}^T$. The fuzzy subset of $V_{structure}^T$ is defined by a membership function that associated with each element a of $V_{structure}^T$, the extent (between 0 and 1) to which a is a member of this subset: $\mu_V: V_{structure}^T \rightarrow [0, 1]$. In the presence of a rule in format $[\alpha_{ij}] \rightarrow [\beta_{ij}]$, if $[\beta_{ij}]$ is characterized by the membership function $\mu_\beta = \min(\mu_{v_1}, \mu_{v_2} \dots \mu_{v_j} \dots \mu_{v_n})$, and if the rule $[\alpha_{ij}] \rightarrow [\beta_{ij}]$ is characterized by the membership function $\mu_{\alpha \rightarrow \beta}$, then $[\alpha_{ij}]$ is defined by the membership function $\mu_\alpha = \min(\mu_\beta, \mu_{\alpha \rightarrow \beta})$. In this way we can determine the membership function of each nonterminal, and therefore, of axiom S . The feature grammar that benefits from these characteristics is called the fuzzy feature grammar.

2.3. Application

Given a set of features (Fig. 2)

$$X = \left\{ \begin{array}{l} \text{Step, Slot, Blind Slot, Pocket, Blind Step,} \\ \text{Simple Blind Slot, Partial Hole, Hole} \end{array} \right\}$$

A feature can be represented by the selected topological and geometric entity graph, defined as follows:

DEFINITION. For two given sets

$$F^* = \{(f_i, e_i^*) | f_i \in F\}$$

where $F = \{f_1, f_2 \dots f_i \dots f_m\}$ is a set of faces; $e_i^* = (a_2^{\text{tp}}, a_3^{\text{geo}})$ is an entity associated with each face f_i (see Table 1);

$$E = \{(f_i, f_j, e_{ij}) | f_i, f_j \in F\}$$

where $e_{ij} = (\text{adjacents, concave}, a_2^{\text{geo}})$ (see Table 1), we call $G = (F^*, E)$ the selected topological and geometric entity graph. ■

For example, the representation of <Slot> and <Simple Blind Slot> features by the selected topological and geometric entity graph is illustrated below (Fig. 3).

The strength of feature grammars for feature representation will depend on terminal vocabulary definition. For terminals definition, we are using the following hypothesis: a terminal represents a robust connection between faces of features. The problem of robust connection in the selected topological and geometric entity graph can be seen as the obtaining of the full maximal subgraphs of this graph. It is equivalent to decomposition of one similarity relation into maximal similarity subrelations. Considering the set of features X, we have two terminals (Fig. 4).

The relationship between the set of features and the set of terminals can be represented by the bipartite graph (Fig. 5). The graph decomposition yields two classes of features: $C_1^X = \{\text{Step, Slot, Partial Hole, Hole}\}$ and $C_2^X = \{\text{Blind Step, Simple Blind Slot, Blind Slot, Pocket}\}$, corresponding to two classes of terminals: $C_1^Y = \{a\}$ and $C_2^Y = \{b\}$.

Two fuzzy feature grammars $G_{Feature}^C$ are inferred for the feature classes: $C_1^X = \{\text{Step, Slot, Partial Hole, Hole}\}$ and

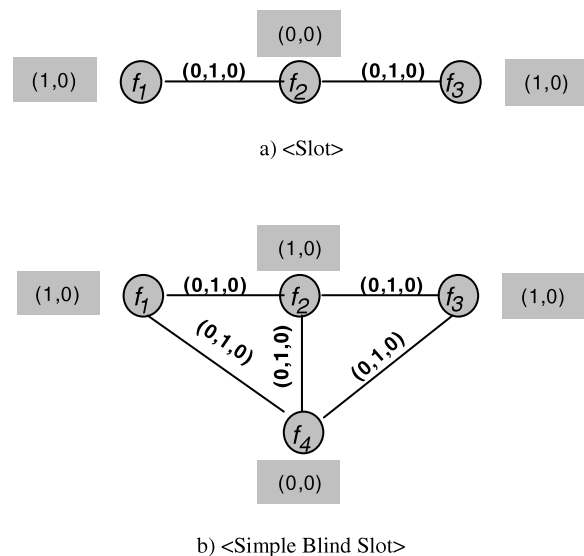


Fig. 3. Selected topological and geometric entity graph.

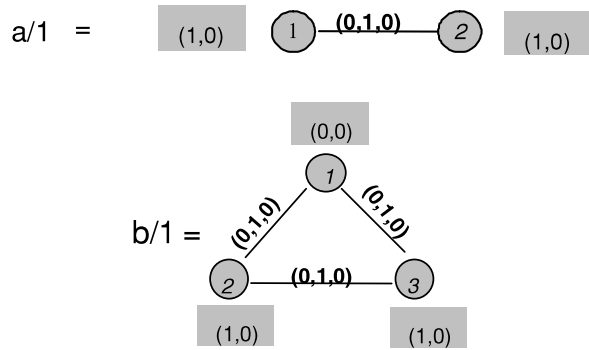


Fig. 4. Selected topological and geometric entity graph.

$C_2^X = \{\text{Blind Step, Simple Blind Slot, Blind Slot, Pocket}\}$, respectively. For the first feature class, we have

$$G_{\text{Feature}} = \{V_{\text{structure}}^T, V_{\text{joint-tie}}^T, V_{\text{structure}}^N, V_{\text{joint-tie}}^N, S, \nabla, \Lambda, \},$$

where

$$V_{\text{structure}}^T = \{a/\mu_a\},$$

$$V_{\text{joint-tie}}^T = \{0, 1, 2\} \quad m \in N,$$

$$V_{\text{structure}}^N = \left\{ \begin{array}{l} D/\mu_D, E/\mu_E, F/\mu_F, \text{Step}/\mu_{\text{Step}}, \\ \text{Slot}/\mu_{\text{Slot}}, \text{Partial Hole}/\mu_{\text{Partial Hole}}, \\ \text{Hole}/\mu_{\text{Hole}}, \text{Feature}/\mu_{\text{Feature}} \end{array} \right\},$$

$$V_{\text{joint-tie}}^N = \{O, I, II, \nabla, \Lambda\};$$

$$S = \text{Feature}/\mu_{\text{Feature}}, \nabla, \Lambda;$$

$$P_0 = \begin{bmatrix} [\text{Feature}/\mu_{\text{Feature}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \rightarrow \begin{bmatrix} [\text{Step}/\mu_{\text{Step}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \Big| \begin{bmatrix} [\text{Slot}/\mu_{\text{Slot}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \Big| \begin{bmatrix} [\text{Partial Hole}/\mu_{\text{Partial Hole}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \Big| \begin{bmatrix} [\text{Hole}/\mu_{\text{Hole}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix},$$

$$P_1 = \begin{bmatrix} [\text{Hole}/\mu_{\text{Hole}}] \\ [\nabla] \\ [\nabla] \\ [\Lambda] \\ [\Lambda] \end{bmatrix} \rightarrow \begin{bmatrix} [a/\mu_a \quad E/\mu_E] \\ [1 \quad II] \\ [2 \quad I] \\ [1 \quad II] \\ [2 \quad I] \end{bmatrix} \Big|$$

$$\begin{bmatrix} [a/\mu_a \quad F/\mu_F] \\ [1 \quad II] \\ [2 \quad I] \\ [1 \quad II] \\ [2 \quad I] \end{bmatrix},$$

$$P_2 = \begin{bmatrix} [\text{Partial Hole}/\mu_{\text{Partial Hole}}] \\ [\emptyset] \\ [\Lambda] \\ [\Lambda] \end{bmatrix} \rightarrow \begin{bmatrix} [F/\mu_F] \\ [\emptyset] \\ [I] \\ [II] \end{bmatrix},$$

$$P_3 = \begin{bmatrix} [\text{Slot}/\mu_{\text{Slot}}] \\ [\emptyset] \\ [\Lambda] \\ [\Lambda] \end{bmatrix} \rightarrow \begin{bmatrix} [E/\mu_E] \\ [\emptyset] \\ [I] \\ [II] \end{bmatrix},$$

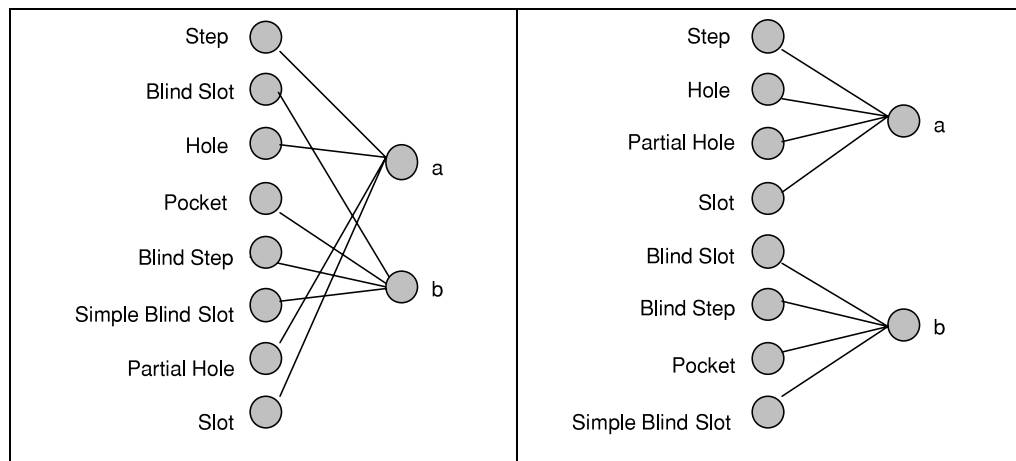


Fig. 5. The bipartite graph and its decomposition.

$$P_4 = \begin{bmatrix} [\text{Step}/\mu_{\text{Step}}] \\ [\emptyset] \\ \Lambda \\ \Lambda \end{bmatrix} \rightarrow \begin{bmatrix} [D/\mu_D] \\ [\emptyset] \\ \text{I} \\ \text{II} \end{bmatrix},$$

$$P_5 = \begin{bmatrix} [F/\mu_F] \\ [\nabla] \\ \text{I} \\ \text{II} \end{bmatrix} \rightarrow \begin{bmatrix} [a/\mu_a \ E/\mu_E] \\ [2 \ 1] \\ [1 \ 0] \\ [0 \ \text{II}] \end{bmatrix} \parallel \begin{bmatrix} [a/\mu_a \ F/\mu_F] \\ [2 \ \text{I}] \\ [1 \ 0] \\ [0 \ \text{II}] \end{bmatrix},$$

$$P_6 = \begin{bmatrix} [E/\mu_E] \\ [\nabla] \\ \text{I} \\ \text{II} \end{bmatrix} \rightarrow \begin{bmatrix} [a/\mu_a \ D/\mu_D] \\ [2 \ \text{I}] \\ [1 \ 0] \\ [0 \ \text{II}] \end{bmatrix},$$

$$P_7 = \begin{bmatrix} [D/\mu_D] \\ [\emptyset] \\ \text{I} \\ \text{II} \end{bmatrix} \rightarrow \begin{bmatrix} [a/\mu_a] \\ [\emptyset] \\ [1] \\ [2] \end{bmatrix}.$$

The fuzzy feature grammar represents the syntactic and fuzzy aspect of features. Structures with the same syntax may represent features with different semantics. Thus, we can build the conditional and fuzzy feature grammar. In this case, the first level of production rules will be associated by conditions. For example, for the first level of production rules P_6 , we have the following semantic condition:

Structures b and A (on the right side of rule $B \rightarrow bA$) are attached if the direction of the main vector A (on the right side) is the same as the direction of the vector $\vec{n}_1 \wedge \vec{n}_2$ of b , where 1 and 2 represent the attaching elements of b .

The previous condition is used in a similar fashion for rules P_1 , P_3 , and P_5 . We will have the following condition for the first level of production rules:

The direction of the main vector of A (left side of the rule $A \rightarrow b$) is initialized from $\vec{n}_1 \wedge \vec{n}_2$ of b , where 1 and 2 represent the attaching elements of b .

There are no semantic conditions to be satisfied for the other rules.

For the second class, we have

$$G_{\text{Feature}}^C = \{V_{\text{structure}}^T, V_{\text{joint-tie}}^T, V_{\text{structure}}^N, V_{\text{joint-tie}}^N, S, \nabla, \Lambda, P\},$$

where

$$V_{\text{structure}}^T = \{b/\mu_b\},$$

$$V_{\text{joint-tie}}^T = \{0, 1, 2, 3\} \quad m \in N,$$

$$V_{\text{structure}}^N = \left\{ \begin{array}{l} A/\mu_A, B/\mu_B, \text{Blind Step}/\mu_{\text{Blind Step}}, \\ \text{Simple Blind Slot}/\mu_{\text{Simple Blind Slot}}, \\ \text{Blind Slot}/\mu_{\text{Blind Slot}}, \text{Pocket}/\mu_{\text{Pocket}}, \\ \text{Feature}/\mu_{\text{Feature}} \end{array} \right\},$$

$$V_{\text{joint-tie}}^N = \{O, \text{I}, \text{II}, \text{III}, \nabla, \Lambda\};$$

$$S = \text{Feature}/\mu_{\text{Feature}}, \nabla, \Lambda;$$

$$P_0 = \begin{bmatrix} [\text{Feature}/\mu_F] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \rightarrow \begin{bmatrix} [\text{Blind Step}/\mu_{\text{Blind Step}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \parallel$$

$$\begin{bmatrix} [\text{Simple Blind Slot}/\mu_{\text{Simple Blind Slot}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \parallel$$

$$\begin{bmatrix} [\text{Blind Slot}/\mu_{\text{Blind Slot}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix} \parallel \begin{bmatrix} [\text{Pocket}/\mu_{\text{Pocket}}] \\ [\emptyset] \\ [\emptyset] \end{bmatrix},$$

$$P_1 = \begin{bmatrix} [\text{Pocket}/\mu_{\text{Pocket}}] \\ \nabla \\ \nabla \\ \nabla \\ \Lambda \\ \Lambda \\ \Lambda \end{bmatrix} \rightarrow \begin{bmatrix} [b/\mu_b \ B/\mu_B] \\ [1 \ \text{I}] \\ [2 \ \text{III}] \\ [3 \ \text{II}] \\ [1 \ \text{I}] \\ [2 \ \text{III}] \\ [3 \ \text{II}] \end{bmatrix} \parallel$$

$$\begin{bmatrix} [b/\mu_b \ C/\mu_C] \\ [1 \ \text{I}] \\ [2 \ \text{III}] \\ [3 \ \text{II}] \\ [1 \ \text{I}] \\ [2 \ \text{III}] \\ [3 \ \text{II}] \end{bmatrix},$$

$$P_2 = \begin{bmatrix} [\text{Blind Spot}/\mu_{\text{Blind Spot}}] \\ [\emptyset] \\ \Lambda \\ \Lambda \\ \Lambda \end{bmatrix} \rightarrow \begin{bmatrix} [C/\mu_C] \\ [\emptyset] \\ \text{I} \\ \text{II} \\ \text{III} \end{bmatrix},$$

$$P_3 = \begin{bmatrix} [\text{Simple Blind Slot}/\mu_{\text{Simple Blind Slot}}] \\ [\emptyset] \\ \Lambda \\ \Lambda \\ \Lambda \end{bmatrix} \rightarrow \begin{bmatrix} [B/\mu_B] \\ [\emptyset] \\ \text{I} \\ \text{II} \\ \text{III} \end{bmatrix},$$

$$P_4 = \begin{bmatrix} [\text{Blind Step}/\mu_{\text{Blind Step}}] \\ [\emptyset] \\ \Lambda \\ \Lambda \\ \Lambda \end{bmatrix} \rightarrow \begin{bmatrix} [A/\mu_A] \\ [\emptyset] \\ \text{I} \\ \text{II} \\ \text{III} \end{bmatrix},$$

$$P_5 = \begin{bmatrix} [C/\mu_B] \\ [I] \\ [\nabla] \\ [I] \\ [II] \\ [III] \end{bmatrix} \rightarrow \begin{bmatrix} [b/\mu_b \ B/\mu_A] \\ [1 \ I] \\ [3 \ II] \\ [1 \ I] \\ [2 \ 0] \\ [0 \ III] \end{bmatrix} \mid \begin{bmatrix} [b/\mu_b \ C/\mu_C] \\ [1 \ I] \\ [3 \ II] \\ [1 \ I] \\ [2 \ 0] \\ [0 \ III] \end{bmatrix},$$

$$P_6 = \begin{bmatrix} [B/\mu_B] \\ [I] \\ [\nabla] \\ [I] \\ [II] \\ [III] \end{bmatrix} \rightarrow \begin{bmatrix} [b/\mu_b \ A/\mu_A] \\ [1 \ I] \\ [3 \ II] \\ [1 \ I] \\ [2 \ 0] \\ [0 \ III] \end{bmatrix},$$

$$P_7 = \begin{bmatrix} [A/\mu_A] \\ [\emptyset] \\ [I] \\ [II] \\ [III] \end{bmatrix} \rightarrow \begin{bmatrix} [b/\mu_b] \\ [\emptyset] \\ [I] \\ [II] \\ [III] \end{bmatrix}.$$

The preceding production rules are associated by conditions similar to the previous class. In the case of this feature class, the mixed product $\vec{n}_1 \cdot (\vec{n}_2 \wedge \vec{n}_3)$ is considered. Thus, the structures (in the right part) are attached if the sign of the mixed product $\vec{n}_1 \cdot (\vec{n}_2 \wedge \vec{n}_3)$ does not change.

3. FEATURE RECOGNITION APPROACH

Using the principles discussed above, we have developed a new feature recognition approach. The flow chart in Figure 6 shows the main phases of this approach. The first phase, called *regioning*, consists of identifying the potential zones for the birth of features. The second phase, called *virtual extension*, consists of building links and virtual faces. The third phase, called *structuring*, consists of transforming the region into a structure compatible with the structure of the features represented by the feature grammar. The fourth phase, called *identification*, consists of identifying the features in these zones. The fifth phase, called, *modeling*, consists of representing the model either by regions or by features.

3.1. Regioning

A *region* defines a potential area of a part where either canonical features or features in interaction may be recognized. During the interaction, features may have lost their concavity. As a result, some faces of features in interaction are not identified during the recognition phase. In this case, the potential region for feature recognition is expanded with concave border faces (*local expansion principle*). Thus, we can define a

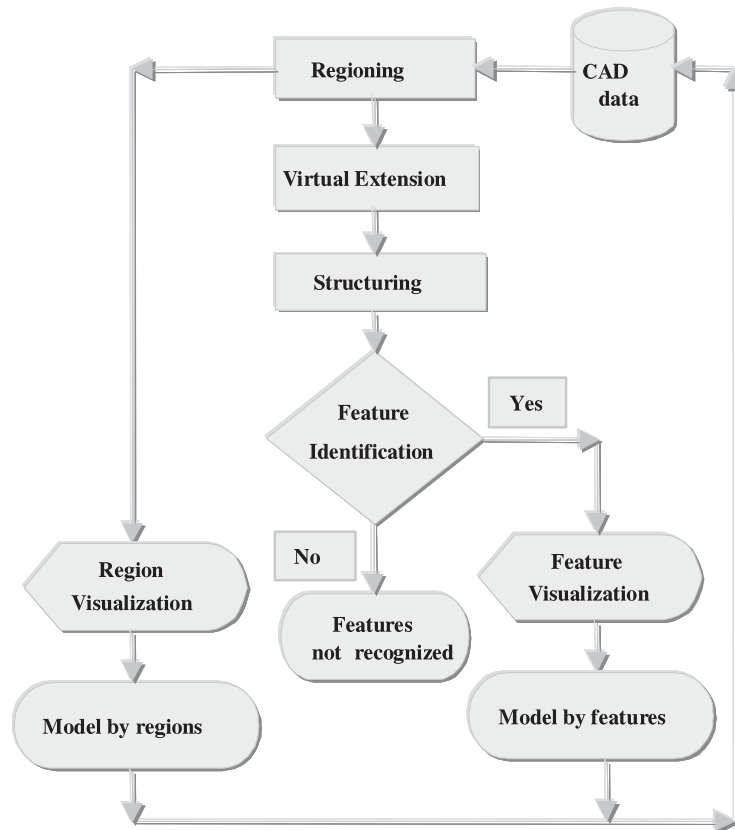


Fig. 6. A flow chart of the feature recognition approach.

region of the part as a set of connected faces characterized by their concavity or their convexity that may be transformed into concavity. Furthermore, the interaction between features may produce neighboring regions that may be either adjacent or recoverable. If $\{R_1 \dots R_i, R_{i+1} \dots R_n\}$ is a set of regions, then a macroregion R may be defined by grouping a set of neighboring regions (*global expansion principle*). For example, the part shown below (Fig. 7a) contains two regions. The first region comprises concave faces 1, 2, and 3 and convex face 7, which may be transformed to concave by the virtual extension toward face 1. The second region comprises concave faces 4, 5, and 6 and convex face 7. In this case, the convex face 7 can be transformed to concave by virtual extension toward face 6. These two regions share face 7. As a result, a macroregion is defined by $\langle \text{macroregion}_1 \rangle \rightarrow \langle \text{region}_{11} \rangle \langle \text{region}_{12} \rangle$, where $\langle \text{region}_{11} \rangle$ and $\langle \text{region}_{12} \rangle$ are the first and second regions, respectively, of the first macroregion $\langle \text{macroregion}_1 \rangle$. The part in Marefat and Kashyap (1990; Fig. 7b) comprises a region that includes concave faces 1, 2, 3, 4, 5, and 6.

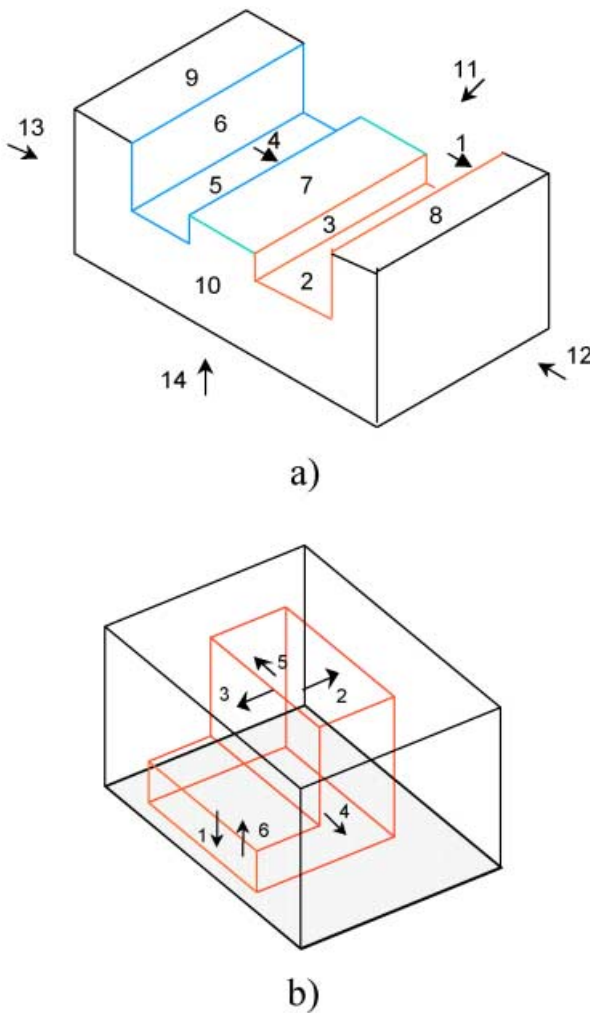


Fig. 7. Examples of regions. [A color version of this figure can be viewed online at www.journals.cambridge.org]

Thus, the regioning procedure involves three subphases:

Subphase 1: Search for regions. The faces characterized by concavity are called the *primary faces* of the region. This phase consists of extracting the set of primary faces from the part and grouping them into regions.

Subphase 2: Local expansion. Based on the *local expansion principle*, the border faces, called the *secondary faces*, characterized by their convexity susceptible to be transformed into concavity, are assigned to the region in question.

Subphase 3: Creating macroregions. Based on the *global expansion principle*, regions are grouped together into macroregions.

EXAMPLE 1. Given two parts (Fig. 7a, b). For the part in Figure 7a we can write the following:

$$\langle \text{part} \rangle \rightarrow \langle \text{macroregion}_1 \rangle, \tag{10}$$

$$\langle \text{macroregion}_1 \rangle \rightarrow \langle \text{region}_{11} \rangle \langle \text{region}_{12} \rangle, \tag{11}$$

$$\langle \text{region}_{11} \rangle \rightarrow \langle \text{primary faces}_{11} \rangle \langle \text{secondary faces}_{11} \rangle,$$

$$\langle \text{region}_{12} \rangle \rightarrow \langle \text{primary faces}_{12} \rangle \langle \text{secondary faces}_{12} \rangle, \tag{12}$$

$$\langle \text{primary faces}_{11} \rangle \rightarrow \langle f_1 \rangle \langle f_2 \rangle \langle f_3 \rangle,$$

$$\langle \text{primary faces}_{12} \rangle \rightarrow \langle f_4 \rangle \langle f_5 \rangle \langle f_6 \rangle, \tag{13}$$

$$\langle \text{secondary faces}_{11} \rangle \rightarrow \langle f_7 \rangle,$$

$$\langle \text{secondary faces}_{12} \rangle \rightarrow \langle f_7 \rangle. \tag{14}$$

■

3.2. Virtual extension

The faces in a region can be divided into three classes:

- class 1:* primary faces characterized by concavity only;
- class 2:* secondary faces characterized by convexity only;
- and
- class 3:* primary faces characterized by both concavity and convexity.

The first class concerns faces that resist to interaction. For example, faces 1, 2, and 3 and 4, 5, and 6 in the part in Figure 7a and faces 1, 2, 4, and 5 of the part in Figure 7b belong to this class. Despite the interaction, they kept their concavity characteristic. The second class concerns border faces. These convex faces probably lost all of their original concavity characteristics during the interaction. For example, face 7 (Fig. 7a) belongs to this class. The third class concerns primary faces, which probably lost their concavity characteristic during the interaction. For example, faces

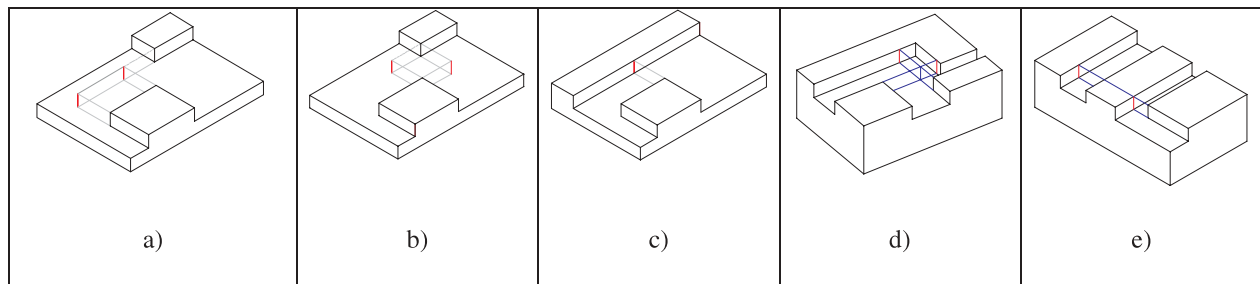


Fig. 8. Illustrations of conditions. [A color version of this figure can be viewed online at www.journals.cambridge.org]

3 and 6 (Fig. 7b) belong to this class. Thus, virtual extension consists of transforming the convex faces in the second and third classes into concave faces by generating virtual links. These links must meet the following conditions:

Condition 1: A pair of virtually extended faces belongs to a group of extended faces, if and only if their virtual adjacency is concave (Fig. 8a).

Condition 2: Given two convex edges e_i and e_j , faces X and Y are adjacent in e_i and faces Z and V are adjacent in e_j . If the virtual extension of faces X and Y and the virtual extension of faces Z and V create two edges, called virtual–virtual edges, then those edges are considered simultaneously (Fig. 8b).

Condition 3: If condition 2 is false, then between two adjacent convex faces, one and only one face can be virtually extended by forming an edge, called a virtual–real edge. This edge will jointly belong to the extended virtual face and the real virtually intersected face (Fig. 8c).

Condition 4: In a set, if each of the virtually extended faces form virtual–real edges, and if the virtual extensions intersect, then the selection of one of those faces penalizes the others (Fig. 8d).

Condition 5: The virtually extended face does not intersect the interior parts of the part faces (Fig. 8e).

The recomposition of the virtual face is a special case for the generation of virtual links. The interaction between the canonical features may break a face down into a group of *small faces*. Then those small faces may be unified into a virtual face. The minimal conditions for a group of faces to be unified are as follows (Marefat & Kashyap, 1990):

condition 1: the faces must have the same support surface, that is, the same equation;

condition 2: the normals of the faces must have the same directions; and

condition 3: the unified face must not intersect the interior parts of the part faces.

The first and second conditions show that the faces have the same geometry, while the third shows that the unified face cannot be destroyed by the other faces of the part. We define an order relation between the generation of virtual faces and virtual links: *we first try to generate the virtual faces by unifying the groups of faces that meet the minimal conditions. If a virtual face is generated from the unification of a group of faces, then any face in that group must be virtually extended to form virtual links.*

EXAMPLE 2. Given the parts in Figure 7a and b. For the part in Figure 7a, face 7 may be transformed into a concave face by generating virtual links with faces 6 and 1 matrix (Fig. 9a), while for the part in Figure 7b, face 3 and face 6 may be transformed into concave faces by generating virtual links with faces 1 and 2, respectively (Fig. 9b). ■

| | | Faces | | | | | | |
|-------|---|-------|-------|-------|-------|-------|-------|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Faces | 1 | | 0.1.0 | | | | | |
| | 2 | 0.1.0 | | 0.1.0 | | | | |
| | 3 | | 0.1.0 | | | | | |
| | 4 | | | | | 0.1.0 | | |
| | 5 | | | | 0.1.0 | | 0.1.0 | |
| | 6 | | | | | 0.1.0 | | |
| | 7 | 3.1.0 | | | | | | 3.1.0 |

a)

| | | Faces | | | | | |
|-------|---|-------|-------|-------|-------|-------|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| Faces | 1 | | 0.1.0 | | 0.1.0 | 0.1.0 | |
| | 2 | 0.1.0 | | | 0.1.0 | 0.1.0 | |
| | 3 | 3.1.0 | | | 0.1.0 | 0.1.0 | |
| | 4 | 0.1.0 | 0.1.0 | 0.1.0 | | | 0.1.0 |
| | 5 | 0.1.0 | 0.1.0 | 0.1.0 | 0.1.0 | | 0.1.0 |
| | 6 | | 3.1.0 | | 0.1.0 | 0.1.0 | |

b)

Fig. 9. A matrix representing the real or virtual concavity between faces.

3.3. Structuring

Structuring consists of converting the representation of macroregions (created by regioning) with virtual faces and links (created by virtual extension) into a structure compatible with the feature grammar definition. Structuring involves the following subphases described in Section 3:

subphase 1: searching for terminals and

subphase 2: creating canonical matrices (terminal matrix and joint matrix).

EXAMPLE 3. Given the part in Figure 7a and its matrix representing the relation between the faces (Fig. 9a). The terminals found and the canonical matrix are shown in Figure 10a and b.

These terminals are type *a*. The terminal with faces 4 and 5 is colored first. Face 4 is colored with color 1 and face 5 with color 2. The terminal with faces 5 and 6 is colored second. As face 5 is colored with color 2 in a previous terminal, in this terminal it is colored with color 1. As a result, face 6 is colored with color 2. The membership function $\mu_V(a)$ associated with each terminal shows the extent to which that terminal is similar to terminal *a*. The terminal with a virtual concave edge is associated with $\mu_V(a) = 0.8$. ■

EXAMPLE 4. Given the part in Figure 7b and its matrix representing the relation between the faces (Fig. 9b). The terminals found and the canonical matrix are shown in Figure 11.

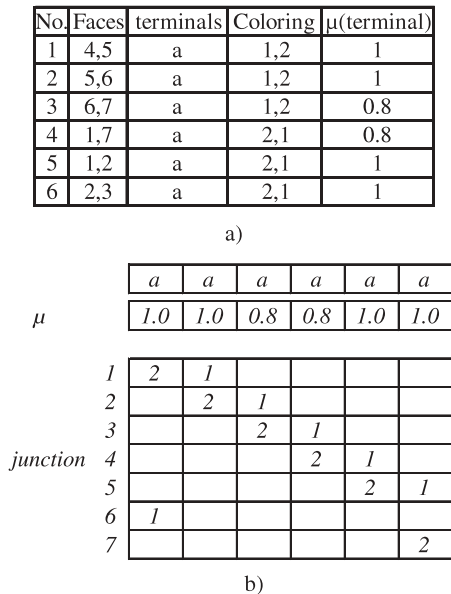


Fig. 10. The terminals and the canonical matrix of the recognized macroregion.

| No. | Faces | terminals | Coloring | $\mu(\text{terminal})$ |
|-----|-------|-----------|----------|------------------------|
| 1 | 2,1,4 | b | 2,1,3 | 1 |
| 2 | 2,1,5 | b | 3,1,2 | 1 |
| 3 | 5,1,3 | b | 3,1,2 | 0.8 |
| 4 | 3,1,4 | b | 3,1,2 | 0.8 |
| 5 | 5,6,2 | b | * | 0.3 |
| 6 | 2,6,4 | b | * | 0.3 |

a)

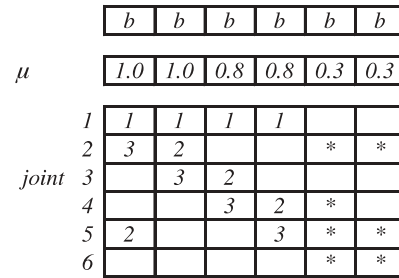


Fig. 11. The terminals and the canonical matrix of the recognized macroregion.

The transformation of the maximum subrelations into colored terminals is shown in Figure 9b. These are type *b* terminals. For any terminal, the face considered as the base is colored with color 1. Thus, face 1 is colored with color 1. The other faces are colored using the same procedure. For example, the terminal with faces 1, 2, and 4 is colored first (face 1 is already colored). Face 2 is colored with color 2 and face 4 with color 3. The terminal with faces 1, 2, and 5 is colored second. As face 2 was colored with color 2 in a previous terminal, in this terminal it is colored with color 3. As a result, face 5 is colored with color 3. Thus, we continue coloring for all terminals. Terminals 5 and 6, with faces 6, 5, and 2 and 6, 2, and 4, respectively, are not colored because faces 5, 2, and 4 are already colored with colors 2 or 3. Furthermore, we consider that there is only one face with the characteristic base in a macroregion or a region. The membership function $\mu_V(a)$ associated with each terminal shows the extent to which the terminal is similar to terminal *b*. In the case of noncolored faces, this function takes the value of $\mu_V(b) = 0.3$. The asterisk shows that terminals 5 and 6 participate in forming joints 2, 4, 5, and 6, but those terminals are not colored. ■

3.4. Identification

The representation and the resolution of the recognition problem, for either canonical features or features in interaction, is shown in a state graph. Consider feature grammar *G* and a feature structure to be analyzed, represented by the canonical matrix (terminals matrix and joint matrix). A *state* is any possible rewriting of canonical matrices (terminal matrix and joint matrix) by one or more production rules of

feature grammar G . States will constitute the nodes of the state graph. An arc will connect two states if we can pass from one state to another by a single application of a production rule of fuzzy feature grammar G or if the structure is updated after the recognition of a canonical feature in a region of features in interaction. The structure update consists of the following:

- virtually removing the faces and edges that are exclusively part of the recognized feature and considering the Virtual Extension conditions (*union principle*);
- reusing the faces shared by several features, knowing that if n virtual links are created to a real face, then that face will be shared by $n + 1$ features (*sharing principle*);
- creating new terminals knowing that if a face and/or edges are virtually erased in an order 3 terminal, then the terminal may be transformed into an order 2 terminal(s) (*embedding principle*).

Each arc is labeled with a letter representing either the production rule applied, or the notation for updating the structure. A state can be in one of the following conditions:

Condition 0: It is initial.

Condition 1: It has not been built yet.

Condition 2: It was just built by applying a production rule or by updating the structure.

Condition 3: It is a dead end. Some nodes are dead-end structures, for which no production rules are applicable.

Condition 4: It is terminal. A state is terminal if it represents either the recognized canonical feature, or features in interaction.

By applying the *anchoring principle*, *virtual extension conditions*, *the sharing principle*, *the embedding principle*, and *the union principle*, we have developed the following heuristic for identifying canonical features and/or features in interaction:

STEP 1. The initial structure represented by the canonical matrix defines the initial state (condition 0). The initial state is added to an ordered list of states according to the order relation of their structures.

STEP 2. If the list is empty, then failure; the procedure halts. Otherwise, we choose a state from the list using the following rule: *last entered in the list, first out*. It represents the current state w_i . We remove it from the list and add it to the group of processed states W .

STEP 3. If the current state w_i is a terminal state (condition 4), then success. The production rules can be found using the pointers created in step 5.

STEP 4. If production rules, set in an order relation, can be applied, then we develop the current state w_i by creating the set of successor states Y , otherwise if a feature is recog-

nized, then the structure can be updated (*according to the virtual extension conditions, the sharing principle, the embedding principle, the union principle*) by creating only a successor state in Y (condition 2), otherwise go to 2 (condition 3).

STEP 5. For all states Yw_j , we put w_j in list if it does not already belong there and if it does not belong to the group of processed states. If we put w_j in the list, then we create a pointer w_j to w_i along with the production rule used.

STEP 6. Go to 2.

In the list, *the last* represents a *maximal state*. A state is maximal if it is hierarchically greater than the other states. Between two states at the same hierarchical level, the state from which a new state is formed and that is characterized by a value greater of the membership function μ is considered as a maximal state.

EXAMPLE 5. Given the part in Figure 7a, its canonical matrix (Fig. 10a) and the inferred conditional and fuzzy feature grammar G_{Feature}^C for the feature class $C_1^X = \{\text{Step, Slot, Partial Hole, Hole}\}$. We find three $\langle \text{Slot} \rangle$ features, each of which comprises the following terminals, respectively: 5 and 6 for the first, 1 and 2 for the second, and 3 and 4 for the third. Each of these features is associated with the respective value of the membership function: $\mu = 1$ for the first and second features and $\mu = 0.8$ for the third. ■

EXAMPLE 6. Given the part in Figure 7b, its canonical matrix (Fig. 10b) and the inferred conditional and fuzzy feature grammars G_{Features}^C for the feature classes: $C_1^X = \{\text{Step, Slot, Partial Hole, Hole}\}$ and $C_2^X = \{\text{Blind Step, Simple Blind Slot, Blind Slot, Pocket}\}$. The canonical matrix (Fig. 10b) represents the initial state. By successively applying rules P_7, P_6, P_5, P_1, P_0 (developed on the right) in iterations 1, 2, 3, 4, and 5, we obtain the “Pocket” recognition feature with $\mu_{\text{Pocket}} = 0.8$. This feature comprises faces 1, 2, 3, 4, and 5. The structure is updated in the *sixth iteration*. According to Condition 3 of the virtual extension conditions, the virtual extension of face 3 penalizes the virtual extension of face 6. Thus, virtual edge 6, 2 will no longer exist in the fifth and sixth terminals. According to the embedding principle, the fifth terminal of type *b* comprising faces 2, 5, 6 will be transformed into two type *a* terminals comprising faces 2, 5 and 5, 6, respectively. Similarly, the sixth terminal of type *b* comprising faces 2, 4, 6 will be transformed into two type *a* terminals comprising faces 2, 4 and 6, 4, respectively. As both terminals comprising faces 2, 5 and faces 2, 4, respectively, are already used in the two terminals comprising faces (1, 2, 5) and faces (1, 2, 4), respectively, of the Pocket feature, then they will not be considered for the recognition of a new feature. Thus, the two terminals comprising faces 5, 6 and 6, 4, respectively, remain to be considered. According to the sharing principle, face 1 will be shared with other features because a virtual link is built to it. Faces 2 and 3 will be erased because

these faces do not exist in the remaining terminals. As a result, according to the embedding principle, we will have two terminals comprising faces 1, 5 and 4, 1, respectively. Finally, we have four type *a* terminals, comprising faces (1, 5), (5, 6), (6, 4), (4, 1), respectively. The matrix in Figure 12 shows the updated structure.

By applying in succession rules $P_7, P_6, P_5, P_1,$ and P_0 of the conditional and fuzzy feature grammar $G_{Feature}^C$, inferred for class $C_1^X = \{\text{Step, Slot, Partial Hole, Hole}\}$, we finally recognize the $\langle \text{Hole} \rangle$ feature with $\mu_{\text{Hole}} = 1$. ■

3.5. Modeling

The regioning phase transforms the representation B.Rep of the part into a representation by regions. The feature recognition in the identification phase transforms either the representation by macroregions into a representation by features (macroregions) or the representation by regions into a representation by features (regions). For example, for the part in Figure 7a we have

$$\langle \text{macroregion}_1 \rangle \rightarrow \langle \text{Slot} \rangle \langle \text{Slot} \rangle \langle \text{Slot} \rangle, \tag{15}$$

$$\langle \text{Slot} \rangle \rightarrow \langle f_1 \rangle \langle f_2 \rangle \langle f_3 \rangle,$$

$$\langle \text{Slot} \rangle \rightarrow \langle f_4 \rangle \langle f_5 \rangle \langle f_6 \rangle,$$

$$\langle \text{Slot} \rangle \rightarrow \langle f_1 \rangle \langle f_7 \rangle \langle f_6 \rangle. \tag{16}$$

Equation (15) shows that the macroregion comprises three $\langle \text{Slots} \rangle$; Eq. (16) shows the faces that make up each $\langle \text{Slot} \rangle$.

4. APPLICATION

To give a comparative example, consider the part in Marefat and Kashyap (1990; Fig. 13). Table 2 gives the results of the feature recognition.

If we compare our results with those of Marefat and Kashyap (1990), we can see several differences. For exam-

| | | terminals | | | | |
|-------|--|-----------|---|---|---|---|
| | | a | a | a | a | |
| μ | | 1 | 1 | 1 | 1 | |
| | | 2 | 1 | | | 1 |
| joint | | | 2 | 1 | | 2 |
| | | | | 2 | 1 | 3 |
| | | 1 | | | 2 | 4 |

Fig. 12. The updated structure.

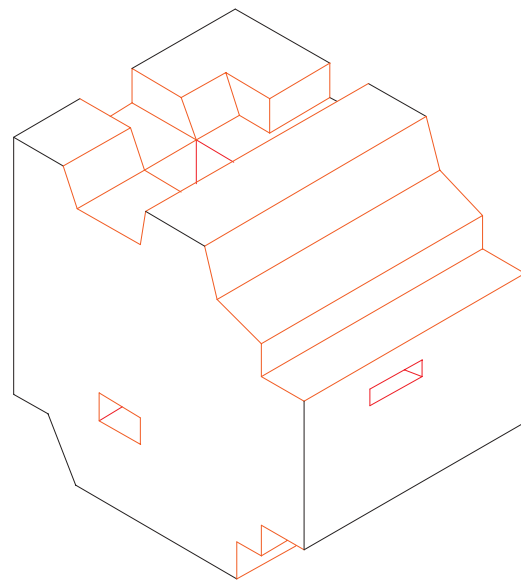
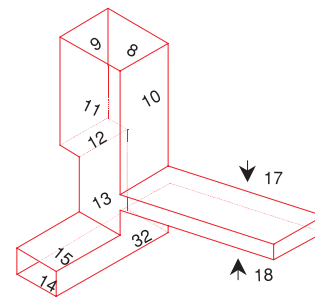
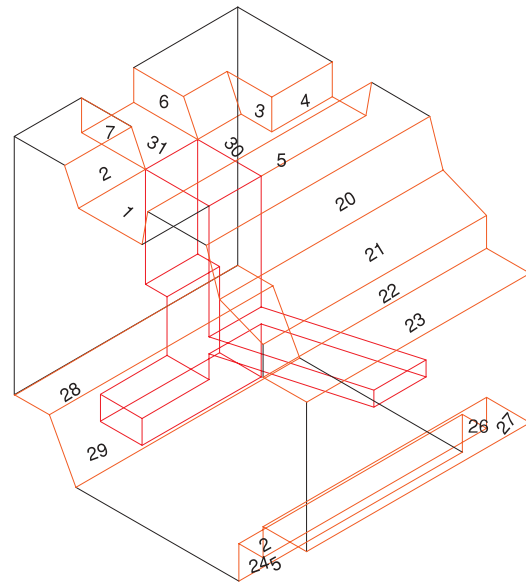


Fig. 13. A part with features. [A color version of this figure can be viewed online at www.journals.cambridge.org]

Table 2. Recognition results

| Macroregions | Regions | Recognized Features | Faces | μ |
|--------------|---------|---------------------|----------------------|-------|
| 1 | 1 | Blind step | 1–30–31, 2–33, 3, 5 | 0.8 |
| | | Slot | 7, 1–30–31, 6 | 0.9 |
| | | Slot | 4, 1–30–31, 5 | 0.9 |
| | 2 | Pocket | 8, 13, 14, 10–32, 15 | 0.8 |
| | | Pocket | 12, 10–32, 11, 9, 8 | 0.8 |
| | | Hole | 10–32, 11, 13, 8 | 0.9 |
| 2 | 1 | Hole | 8, 17, 11, 18 | 1 |
| | | Step | 20, 21 | 1 |
| 3 | 2 | Step | 22, 23 | 1 |
| | | Step | 24, 25 | 1 |
| 4 | 1 | Step | 26, 27 | 1 |
| | | Step | 28, 29 | 1 |

ple, in the second region of the first macroregion, first we recognized the two pockets:

| | | |
|--------|----------------------|-----|
| Pocket | 8, 13, 14, 10–32, 15 | 0.8 |
| Pocket | 12, 10–32, 11, 9, 8 | 0.8 |

and then the two holes:

| | | |
|------|------------------|-----|
| Hole | 10–32, 11, 13, 8 | 0.9 |
| Hole | 8, 17, 11, 18 | 1 |

Thus, our approach implicitly follows the idea of extracting the volume of a recognized feature. In the approach of Marefat and Kashyap (1990), the recognition of the preceding two pockets had no influence on subsequent feature recognitions. Thus, after recognizing those two pockets, in Marefat and Kashyap (1990), another pocket defined by faces 14, 13, 11, 10–32, and 8 was recognized. The feature modeling system software was developed in a CAD environment (CATIA from Dassault Systems on an IBM RS 6000 workstation).

5. CONCLUSIONS

This paper proposed an approach for recognizing either canonical features or features in interaction. The proposed recognition approach included various phases. The first phase, regioning, consisted of identifying the potential zones for the birth of features. The creation of macroregions, as a subphase in regioning, can be used later to define relations between features. The second phase, virtual extension, consisted of building links and virtual faces. Here we used a possibilistic approach, given the problem for recognizing features in interaction. The third phase, structuring, consisted of transforming the region into a structure compatible with the structure of the features represented by the fuzzy feature grammar. The fourth phase, identification,

consisted of identifying the features in these zones. The heuristic attempts to generate a set of possible solutions based on the fuzzy feature grammar. We must stress that the number of solutions can be reduced if we consider a pre-defined knowledge context. The fifth phase, modeling, consisted of representing the model by features. The feature modeling system software using this approach is implemented in the CATIA CAD–CAM environment. The proposed approach provides a global framework for the feature recognition problem. Feature recognition is highly dependent on the representation by the feature grammar. In cases where the features are not formalized by a grammar and as a result are not recognized, the approach proposes potential regions by differentiating macroregions and the constituent regions. Recognized features are used for various applications.

REFERENCES

- Ansaldi, S., de Floriano, L., & Falcadino, B. (1985). An edge–face relational scheme for boundary representation. *Computer Graphics Forum* 4(4), 319–332.
- Bond, A.H., & Chang, K.J. (1988). Feature based process planning for machined parts. *Proc. ASME Int. Computers in Engineering Conf.*, Vol. 1, pp. 571–576, San Diego, CA.
- Bronsvort, W., & Jansen, F.W. (1993). Feature modelling and conversion—key concepts to concurrent engineering. *Computers in Industry* 21, 61–68.
- Choi, B.K., Barabash, M.M., & Anderson, D.C. (1988). Automatic recognition of machined surfaces from a 3-D solid model. *CAD* 20(2), 58–64.
- De Martino, T., & Gianini, F. (1994). The role of feature recognition in the future CAD systems. *Proc. IFIP Int. Conf. Feature Modelling and Recognition in Advanced CAD/CAM Systems*, pp. 343–356, Valenciennes, France.
- Falcidieno, B., & Gianini, F. (1989). Automatic recognition and representation of shape-based features in a geometric modelling system. *Computers Vision, Graphics and Image Proceedings* 48, 92–123.
- Henderson, M.R. (1994). Manufacturing feature identification. *Artificial Neural Networks for Intelligent Manufacturing*, pp. 229–264. London: Chapman & Hall.
- Henderson, M.R., & Anderson, D.C. (1984). Computer recognition and extraction of form features: a CAD/CAM link. *Computers in Industry* 5, 329–339.
- Joshi, S., & Chang, T.S. (1988). Graph based heuristics for recognition of machined features from 3-D solid model. *CAD* 20, 58–66.
- Kim, Y.S. (1992). Recognition of form features using convex decomposition. *CAD* 24(9), 461–476.
- Kim, Y.S., & Wilde, D.J. (1992). A convex decomposition using convex hulls and local cause of its non-convergence. *ASME Journal of Mechanical Design* 114, 459–467.
- Laako, T., & Mäntylä, M. (1993). Feature modelling by incremental feature recognition. *CAD* 25(8), 479–492.
- Li, R.K. (1988). A part-feature recognition system for rotational parts. *International Journal of Product Research* 26(9), 1451–1475.
- Marefat, M., & Kashyap, R.L. (1990). Geometric reasoning for recognition of three dimensional object features. *Transactions of the 8th Army Conf. Applied Mathematics and Computing*, pp. 705–731.
- Menon, S., & Kim, Y.S. (1994). Cylindrical features in form feature recognition using convex decomposition. *Proc. IFIP Int. Conf. Feature Modelling and Recognition in Advanced CAD/CAM Systems*, pp. 295–314, Valenciennes, France.
- Srinivasan, R., Liu, C.R., & Fu, S.K. (1985). Extraction of manufacturing details from Geometric models. *Computers and Industrial Engineering* 9, 125–134.
- Woo, T. (1982). Feature extraction by volume decomposition. *Proc. CAD/CAM Technology in Mechanical Engineering Conf.*, pp. 76–94. Cambridge, MA: MIT.

Egon Ostrosi obtained his PhD from the Université Louis Pasteur de Strasbourg. He is currently an Associate Professor in the Laboratoire de Recherche Mécatronique 3M at Université de Technologie de Belfort–Montbéliard. Dr. Ostrosi's current research concerns integrated mechanical product design and modeling, knowledge engineering, and the application of knowledge engineering in mechanical design and concurrent engineering.

Michel Ferney obtained his PhD from the Université de Franche–Comté. He is presently Head Professor in and Director of the Laboratoire de Recherche Mécatronique 3M at the Université de Technologie de Belfort–Montbéliard. He is also Director of CNAM for the region of Franche–Comté. Dr. Ferney's research field of interest covers mechatronic product design and modeling, production systems integration, and concurrent engineering.