16

Context dependent Recognition and Parametrization of Shape Features

F. Giannini, T. De Martino*°, B. Falcidieno**

**Istituto per la Matematica Applicata - C.N.R.*

via De Marini, 6 - 16149 Genova - ITALY *tel. +39 10 64751* *Fax +39 106475660*

giannini(demartino,falcidieno)@ima.ge.cnr.it

°Fraunhofer Institute für Graphische

Wilhelminentrasse, 7- 64283 Darmstadt - GERMANY

Abstract

Feature technology is now becoming popular in most advanced CAD systems. However, feature recognition processes are still necessary both to derive design feature-based description from traditional geometric representations for re-design purposes and to create feature-based description for driving downstream applications. One of the main limitations in existing automatic feature recognition systems is the inability of identifying simple features when feature faces and edges have been split or destroyed as a consequence of the interaction with other features. In this paper, a method for identifying such situations by means of geometric reasoning is presented. The proposed system is able to recognize not only generic features but also specific instances of application features with their defining parameter relations and values.

Keywords

Geometric reasoning, feature-based modeling, automatic feature recognition.

1 INTRODUCTION

CAD systems are in a transition state from tools for supporting only the geometric definition and description of objects to tools for product development, thus modeling product geometric and functional information.

This evolution has required a different object description in order to combine geometry and functional data. In order to do this, the concept of feature has been introduced (Shah 1991, Wilson and Pratt 1988). At present, feature technology is becoming common for providing the designer with predefined macros, which are usually only parametric volumes frequently used and closer to his way of thinking. This allows the designer to reason in terms of feature

M. J. Pratt et al. (eds.), *Product Modeling for Computer Integrated Design and Manufacture*

parameters that are more related to meaningful functional requirements, without requiring their translation in terms of geometric primitives, thus making easier and faster the design activity. Moreover, it is also easier to evaluate different alternatives of the same product and to design new parts by simply modifying already defined .

An alternative approach to the creation of feature-based model is the automatic feature recognition technique (Henderson 1994, Waco and Kim 1992, Clark and Corney 1994, Fields and Anderson 1994, Sakurai and Gossard 1990, Nau et al. 1993, Vandenbrande and Requicha 1990). It has been mainly proposed at research level, because of its complexity. Design by feature systems do not invalidate the necessity of recognition processes, mainly required by the context dependency of features sets (De Martino and Giannini 1994, Pratt 1993). In fact, for instance, a feature-based description created by the designer can be not suitable for performing production analysis. Moreover, automatic feature recognition is necessary for obtaining feature-based models from traditional CAD representations (CSG or B-rep), in order to design new parts by modification of already defined ones. The re-editing of feature is possible only if the defining feature parameters and relations have been identified.

Unfortunately, most of the proposed recognition systems are not able to derive complete feature information, at least not only for simple cases. The major limits in automatic feature recognition systems proposed in literature usually are the difficulty in identifying simple features when they interact in such a way that feature entities have disappeared and the fact that the recognized features are still geometric features, such as generic depressions or protrusions, and not application features, thus not sufficient for driving application activities. The few systems able to deal with these problems, use specific hard coded rules that require an a priori knowledge of the context and code modification when the application context or the used technology change.

In this paper, an approach for context dependent recognition of features from boundary representation is presented. It also deals with interacting configurations and is able to determine the parametric description of the recognized features. It is based on considerations about the relationships between the presence of both simple and interacting features with the morphological modifications occurring in the object parts. Context dependent information such as the preference of negative and positive features as well as feature libraries are externally specified.

2 FEATURE CLASSIFICATION AND PARAMETRIZATION

The proposed approach for feature recognition and classification is founded on considerations related to the correspondence between morphological object characteristics and the presence of features, possibly interacting.

The present work deals only with features constituted by planar faces, but the presented reasoning can be extended to quadratic surfaces by considering also surface curvature.

Looking at the available feature taxonomies, it can be noticed that most of them have a *wholly concave /wholly convex* shape, i.e. all the faces forming the feature are adjacent each other through only concave (or only convex) edges. Conversely, features having mixed adjacencies can be seen as composed by specific patterns of concave and convex adjacencies.

Thus, features can be identified by analysing single or patterns of morphological characteristics (i.e. concavities and convexities) of the surface of the object.

In addition to this, it can be noticed that there is also a relation between curvature variations and interactions between features. Two different types of interactions can be distinguished depending on the kind of modifications produced on the shape of the involved features. The first type of interaction produces *marginal* changes: only portions of feature faces are deleted, see figure 1.a. While in the second type *structural* changes arise: one or more feature faces are split or merged with faces of the other interacting feature. This is the case of the two intersecting slots in figure 1.b, where a lateral face of the first slot is split into two parts and the bottom faces of the two slots result in a unique face in the final object. While marginal modifications can be handled by recognition processes, structural ones make them very difficult. However, also these interactions cause specific morphological modifications in the object introducing not only concavities, as corresponding to the single features, but adding also some convexities between them, like those defined by {f7.1, f11} and {f7.2,f12}.

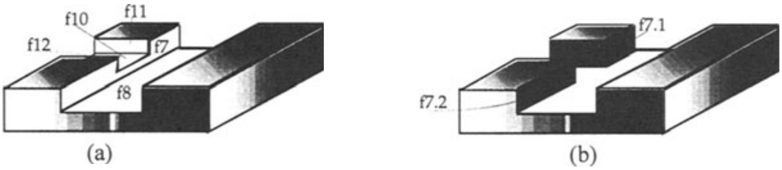


Figure 1 Types of modifications derived by feature interactions: marginal (a) and structural (b)

Based on these considerations, a system able to recognize and parametrize context dependent features, also in interacting situations, is being developed. Such a system acts as a sequence of geometric reasoning and classifications on the object. It takes into consideration the specific context as late as possible thus allowing more flexibility in order to be easier adapted to meet the requirements of different application contexts. The main activities performed by the system are the following, as shown in figure 2:

- ***shape decomposition***: the boundary of the object is decomposed into shape characteristics, i.e. concavities and convexities;
- ***interaction hints classification***: to each shape characteristic a label and a value are assigned indicating the degree of possible presence of an interacting configuration;
- ***shape description reorganisation***: potential missing entities due to feature interaction are created. In this step information regarding the *environment* is used, this indicates the type of features (depressions or protrusions) that have to be privileged. Moreover, a reorganisation of the shape characteristics is performed according to the new created entities;
- ***shape characteristic classification***: each shape characteristic is classified according to its relationships with the rest of the object. Geometric relationships among its faces are determined as well;
- ***context dependent feature classification***: it corresponds to the real context dependent interpretation and requires the feature library to be considered;
- ***parameters association and evaluation***: for each feature the set of the defining parameters, their values and the reference co-ordinate systems are determined.

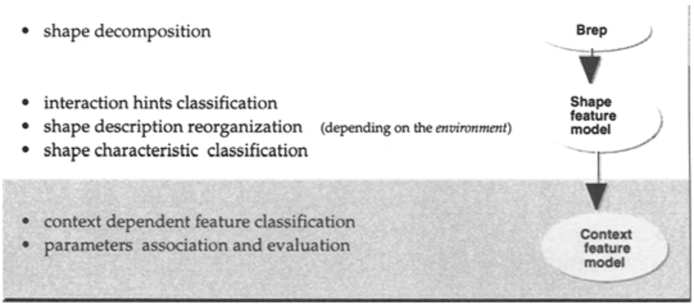


Figure 2 Steps of the recognition system

3 SHAPE DECOMPOSITION

The shape feature recognizer decomposes the object boundary in terms of facesets corresponding to morphological characteristics. The considered characteristics are **weak** and **strong** concavities and convexities, which, for objects bounded by planar faces, are defined as it follows:

a ***weak concavity*** (***convexity***) is a set of connected faces each of them adjacent to *at least another* face in the set through a *concave* (*convex*) edge.

While

a ***strong concavity*** (***convexity***) is a set of faces, each of them adjacent to *all* the other faces of the set through a *concave* (*convex*) edge.

This process is based on a revised version of Kyprianou's approach and uses context independent rules which involve classification of geometric entities (Kyprianou 1980). In fact, the system developed by Kyprianou is able to identify only *weak* shape characteristics, while this system decomposes the object in terms of *strong* shape characteristics, which are much more useful for the identification of interacting situations.

The process starts with the classification of edges and loops as concave or convex, giving rise to an attributed B-rep. Then sets of faces belonging to weak characteristics are identified, thus decomposing the B-rep into facesets. Finally, strong characteristics are recursively and alternately (concavities in convexities and vice versa) searched within the identified facesets. Thus, facesets constituted by only one face can arise, in such cases they maintain their first classification, see figure 3.

The identified facesets are then stored in a hierarchical structure, called Shape Feature Object Graph (SFOG++), (De Martino et al. 1994a). The nodes of this graph correspond to the object shape characteristics and the arcs represent adjacency relationships between the facesets expressed by the shared boundary elements. The direction of the arcs defines the hierarchy of the structure. They are unidirectional, when they express a parent-child relation, or bi-directional, when the connected nodes are at the same level in the hierarchy.

The hierarchy of the model is defined during the shape decomposition step and depends on the type of adjacency between the feature facesets.

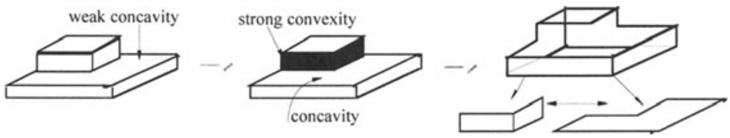


Figure 3 An example of weak and strong characterisation of a simple object and its corresponding SFOG++ representation

4 INTERACTION HINTS CLASSIFICATION

To allow the identification of interacting configurations from shape characteristics, it is necessary to reconstruct entities that have been destroyed due to an interaction, such as portions of edges and faces that have been merged. Usually, there are different potential missing entities, which of them have to be preferred strongly depend on the considered environment, i.e. if only depressions have to be considered or if depressions or protrusions have to be privileged in configurations allowing multiple descriptions, as shown in figure 4.

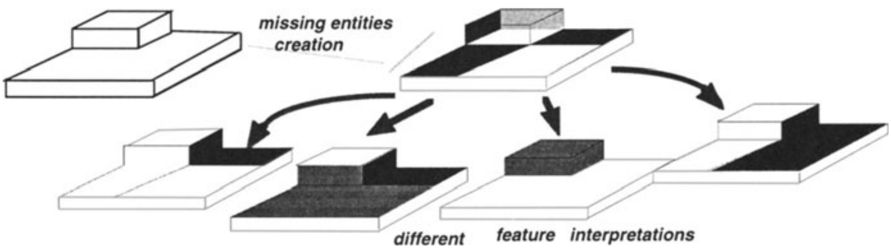


Figure 4 Creation of the possible missing entities for the object in figure 3 and consequent possible feature interpretations

Thus, in order to characterise the environment, three categories of features have been considered. This classification strongly depends on the possible attachment of the features. The considered categories are the following, see figure 5:

- *depressions*: local concavity bounded by convex edges,
- *explicit protrusions*: local convexity bounded by a *closed* concave macro loop,
- *implicit protrusions*: local convexity bounded by an *open* concave macro loop.

The third category corresponds to those protrusions having some missing edges due to the merging of some feature faces with faces of the object in which they are inserted.

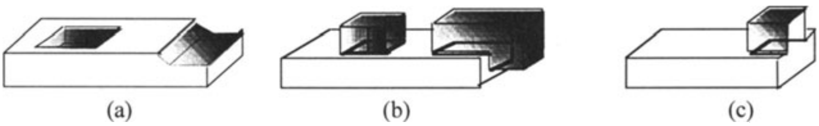


Figure 5 Considered feature categories: (a) depressions, (b) explicit protrusions, (c) implicit protrusions

Thus, considering the fact that the alternate strong classification of concavities and convexities can introduce facesets that are meaningless if considered by themselves, see figure 3, and the possibility that certain contexts are more interested only in some of these categories, to all the facesets of the SFOG++ rep an attribute and a degree of interaction hint are associated (Castanò and Giannini 1995).
The considered labels are the following, see figure 6:

- ***degenerate*** (degree 3) components formed by only one planar face. These are always processed.
  If they are convexities, they correspond to border situations between interacting depressions originating structural changes (i.e. merging of faces). While degenerate concavities indicate the presence of implicit protrusions. Thus, they have also to be further processed in order to create the missing edges and faces, if implicit protrusions are requested, or to derive the dual negative description if depressions are preferred;
- ***implicit convexity*** (degree 2) for convex components whose closed concave macro loops are not fully contained in the weak characteristic from which the convexity has been identified; they may correspond to implicit protrusions;
- ***explicit convexity*** (degree 1) for convex components bounded by a concave macro loop contained in one parent component or in the weak characteristic from which the convexity has been identified, they correspond to explicit protrusions;
- ***disconnected*** (degree 1) for concave components formed by disconnected faces. Also these components are always processed, since they contain interacting features. Depending on the searched features, the corresponding faces will be considered as adjacent, or they will be merged with some adjacent components and re-analysed for searching concavities in concavities.

All the other concavities have degree 0, since they do not give any information related to possible interactions.
The description of how these components are processed to derive missing entities is given in the following paragraph.

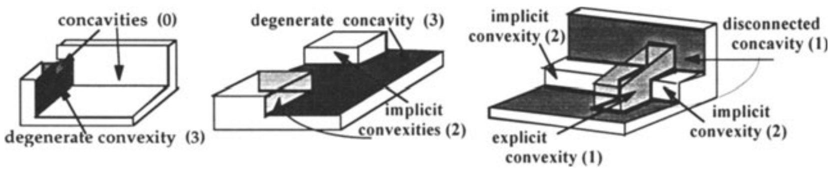


Figure 6 Examples of interaction hint degrees

5 SHAPE DECOMPOSITION RE-ORGANISATION

This step is essential for deriving the facesets corresponding to the features of interest.
The main problem in recognising simple features interaction is related to the fact that some entities have been destroyed, usually edges and vertices, while others, i.e. faces, have been merged with others or split in different faces, thus originating a complete different configuration. Thus, the only possibility for identifying the original simple features is the creation of the missing entities due to the interaction. There are different approaches in doing

this: the first is the maximal approach in which all the possible missing entities are derived, the second is the minimal one in which only the most likely missing entities are created. There are of course advantages and disadvantages in both of them. In the first case, there is the possibility of deriving different feature based descriptions, as shown in figure 4, but it gives rise to explosion problems and requires not easy faceset grouping as well. The second returns only one description by taking into consideration geometric information that strongly characterise specific features, such as parallelism and coplanarity of faces. In this way, the process is faster, and the description returned is easier to be handled. The chosen approach is the second, but we leave also the possibility of deriving all the possible missing entities on request by the user.

In order to derive the most possible feature description, some information derived by the considered *environment* is necessary. This is done by selecting one or a combination of the defined categories. Thus, if the chosen context is machining, only depression features are considered, while for the design context also protrusions are important. Implicit protrusions are used for forcing protrusion description when double descriptions are strongly possible.

Depending on the environment, a subset of interaction hint components is used. Figure 7 shows the correspondence between the selected feature categories and the interaction hints that have to be processed. Thus, for instance, when only depressions have to be considered all the components that are not concavities formed by at least two connected faces (degree 0), have to be re-arranged

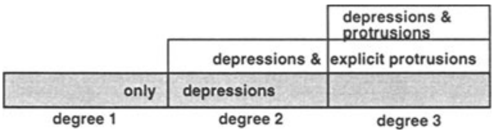


Figure 7 Interaction hints degrees to be processed in correspondence with selected feature category sets.

In general, missing entities are created by intersecting faces belonging to concavities with one or more surfaces selected from those containing faces of the convexities that are concave adjacent to the concavity and have the interaction hint degree to be considered.

The idea is that convexities within concavities have been created by interacting features by cutting and possibly destroying feature entities, as shown in figure 1.b, where the edge between the bottom face and one lateral face of the first slot has been destroyed and substituted with two portions. In this way, the reconstruction of such entities can be done by intersecting the surfaces, on which the faces of the convexities lie, with the faces of the concavity which are concave adjacent. To consider faces means to accept only the pieces of intersection entities that are comprised inside the portion of the surface contained in the boundary of the face. Thus, for the object depicted in figure 1.b, if concavities are preferred, the face to consider is f8, while surfaces have to be selected among those containing f11, f12, f7.1 and f7.2. In order to avoid a heavy segmentation of such faces, not all the surfaces are considered, but only the more promising. To do this, for each concavity, all the convexities to be processed are considered together and geometric relationships between their faces are searched. These relations are: coplanarity and parallelism. Moreover, these faces are sorted according to the number of contained convex edges, convexity degree. Coplanar surfaces are always used, since there are high probabilities that the corresponding faces have been derived from a unique feature face. Then, parallel surfaces and surfaces corresponding to faces with higher convexity degree are used until at least one face for each convex edge has been treated. In figure 8, an example of different entity creation according to the chosen categories is shown.

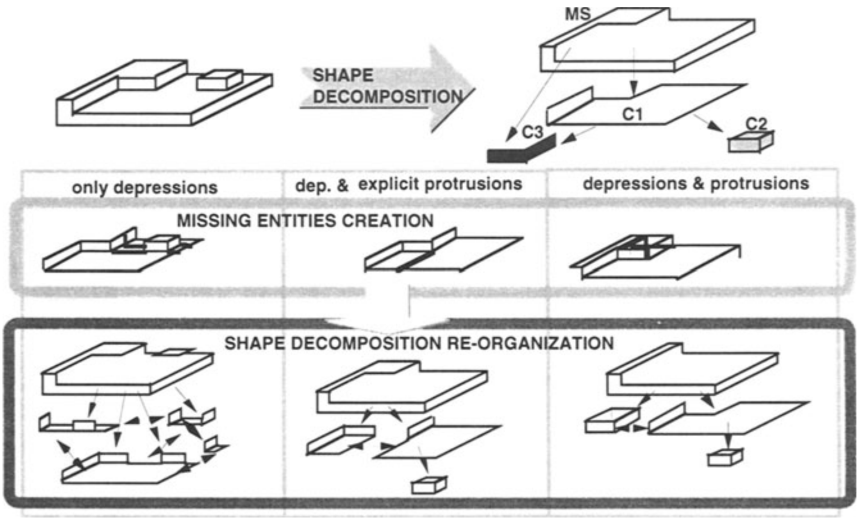


Figure 8 An example of different missing entities creation according to the chosen subset of feature categories.

6 CLASSIFICATION OF THE SHAPE CHARACTERISTICS

Since features are defined by their associated shape and functional meaning, an effective feature recognition system has to deal with both these aspects. Functional meaning can be geometrically translated in terms of the type of interaction between the feature and the object. Thus, feature class and subclass are introduced, where the class is related to the number of object entities modified by the feature, and the subclass considers the changes in the object topology.

Classes and subclasses are evaluated by considering the hierarchical and adjacency relationships between the graph nodes and the so called *connection faces*. A connection face of a feature A is each face of the object which is adjacent to A through at least one edge without belonging to A.

By reasoning on a particular set of connection faces it is possible to evaluate the type of interaction between adjacent face sets and to derive hints to characterise the behaviour of a feature. The hierarchy in the SFOG++ model is used for the identification of the *connection face set* of a feature, defined by all its connection faces belonging to parent or sister features. The connection faces which belong to child components are ignored.

The connection faces of this set are further analysed and distinguished in primary and secondary ones. *Primary connection faces* are those faces which are adjacent to more than one

face of the feature face set while the others are considered *secondary connection faces*. Since primary connection faces interact with a bigger number of feature entities, they are assumed to be more influent on the feature attachment than secondary ones. The information coming from secondary connection faces can be optimised by grouping them into *connection macro-faces*. A connection macro-face corresponds to a connected sequence of secondary connection faces adjacent to a same face of the feature. Thus, the class of a feature A is equal to the number of its primary connection faces plus the number of its connection macro-faces. In this way it is possible to obtain a classification quite independent from the particular shape of the object on which the feature is positioned thus reaching a higher level of abstraction.
In the first slot depicted in figure 9, f_1 and f_5 are primary connection faces since they are adjacent to three faces of the features, while f_2, f_3, f_4, f_6, f_7, f_8 are secondary ones. The secondary connection faces can be considered as two *logical* macro-faces respectively defined by the sets $\{f_2, f_3, f_4\}$ and $\{f_6, f_7, f_8\}$. Thus the faceset A belongs to class 4 as the faceset B.

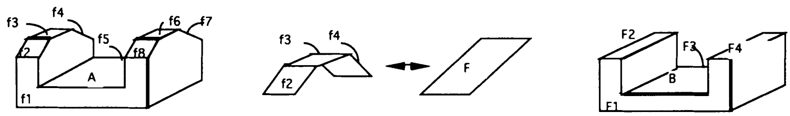


Figure 9 Example of features belonging to the same class

This classification satisfies the requirement to be enough independent of the number of faces of the feature and of the shape of the object, which is interested by the feature. However it does not take into account the topological properties of the features. Thus, another level of specification is needed. Within each class, sub-classes are defined analysing the set of the edges belonging to the boundary of the characteristic faceset and considering the number of connected components of this set. In this way, it is possible to categorise better the functional meaning associated to each faceset, thus distinguishing, for instance, internal through characteristics from blind ones but affecting the same number of connection faces, see figure 10.

Figure 10 Example of feature belonging to the same class but with different subclass

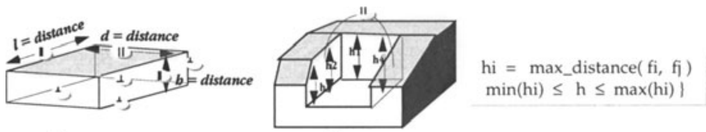
All this information is adequate for identifying generic classes of features sharing the same behaviour but is not sufficient for identifying specific features different in shape. To better describe the shape of the faceset, geometric relationships among the faces of each faceset are determined. The considered geometric relationships are parallelism, coplanarity, angles of incidence.

7 CONTEXT DEPENDENT FEATURE CLASSIFICATION

In this step, information related to the specific context is used. In our system, features in the library are described through their ideal volumes (Pratt 1990) whose shape is described in a declarative way. Faces of the library features are labelled as real or virtual. Real faces are those faces that have to be completely or partially present when the feature is inserted in the object. Virtual faces are generally used for attachment purposes. Feature parameters give the implicit description of the feature volume and are expressed as angle or distance values between both real and virtual entities and are associated to the geometric relations between the referred entities, see figure 11.a.

Information regarding class and sub-class can be directly inserted during the creation of the feature library or can be derived by the number and connectivity analysis of virtual entities. Thus, the number of connected sets of faces correspond to feature sub-class, while their number is related to the feature class (Petta 1994). Such kind of feature representation can be created by using a language or with a teaching by examples technique by creating the feature volume and selecting the virtual entities and inserting the defining parameters.

The classification of the identified facesets is performed by comparing each faceset with the features in the library that have the same type (concavity or convexity) and class and sub-class. A graph-matching is performed on the entities and geometric relationships of the SFOG++ faceset and the real entities and related geometric relations of the feature definition. For implicit protrusions. Once, defined such correspondences, the system performs a matching between the connection faces and the virtual entities. Such correspondence can be one to one or many to one, depending on the shape of the object in which the feature is inserted, see figure 11.b.



(a) (b)

Figure 11 A library feature (a), whose shaded faces are virtual and their correspondent faces in the recognized object (b)

In this way, it is possible to associate all the parameter relations to the facesets. For the parameters involving only real entities, a unique value is returned, the same for virtual entities having one to one correspondence with the connection faces. In the other cases (see figure 11) a range of possible values is returned, and the choice of which value has to be considered is in charge of the user or of the application that will use the created feature based description.

At this point, the associated coordinate systems can be derived.

Unfortunately, there are some cases in which shape characteristics have not the same class of their correspondent feature in the library. Such configurations are usually due to particular feature positions that determine modifications of the real faces of the feature (see figure 12.). For instance, the slot in figure 12.a has been positioned in such a way that the real faces of the slot feature description have been modified. This results that the identified shape feature faceset belongs to class 5.1, while the library slot belongs to class 4.1. A similar hidden situation is given by the faceset corresponding to the pocket in figure 12.b, it belongs to class 4.1 instead of class 1.1. In this case the real lateral faces have been partially removed by the addition of the step. In order to be able to classify also these specific situations, unclassified shape characteristics are compared to features in the library having a lower class value.

Finally, a search for the presence of mixed *curvature* features is performed, by looking for specific patterns of characteristics having homogeneous *curvature*.

(a) (b)

Figure 12 Recognized features having a different class with respect to their corresponding features in the library

8 CONCLUSIONS

In this paper an approach to context dependent feature recognition and parametrization has been shown.
The most important advantages of the approach are:

- possibility of recognising simple features in interacting situations. This is done by adding the most promising missing entities. Upon user request, the system can derive all the possible missing entities in order to allow different feature-based descriptions. This last situation requires additional interpretation process for grouping the derived facesets;
- flexibility. The system does not use any hard coded rule derived by the application. It is mainly based on geometric reasoning, context dependent feature libraries are used only for classification and grouping purposes at the final stage. Moreover, they can be input to the system through a teaching by examples technique, thus easy to be extended and adapted;
- possibility of editing features. The parametric description returned by the system allows evaluation and simulations but also re-editing activities. In fact, knowing the feature type, it is possible to derive the corresponding volume and consequently modify the feature in a design by feature approach.

The presented system is being under development within a co-operation project between the Institute for the Applied Mathematics of the C.N.R., in Genoa, and the Fraunhofer Institute für Graphische, IGD, in Darmstadt, for the development of an modelling system which integrates both design recognition and design by features approach (De Martino et al. 1994b). The system is used both for creating design feature-based representation from traditional boundary descriptions, and for deriving application feature-based representation.

ACKNOWLEDGEMENT

This work is partially supported by the CE within the Human Capital & Mobility Program, Project N. ERBCHBGCT930380. The authors thank their colleagues at I.M.A. and at I.G.D. for the precious discussions on the topic. A special thank to Nicola Castanò, Federico Gamba and Fabio Petta for the implementation and the useful advise.

REFERENCES

Clark, D.E.R., Corney, J, (1994) *Identification of General Protrusion and Depression Features*, *Computer in Engineering*, 1994 Vol. 1 ASME, pp.55-66

Castanò N., Giannini F. (1995),Completamento Volumetrico di caratteristiche di forme in Modelli Boundary, I.M.A. *Technical report* N°15-95

De Martino, T.,Giannini, F., (1994) The role of feature recognition in future CAD systems , Proc. IFIP Int. *Conference on Feature Modeling and Recognition in advanced CAD/CAM Systems*, Valenciennes, May 1994, pp. 343-356

DeMartino T., Falcidieno B., Gamba F., Giannini F., (1994a) Feature Kernel Model: a Reference Model for Integrated Feature based Modeling Systems, *A.I.C.A.* '94, Palermo September 21-23 1994, pp. 1467-1478

De Martino, T., Falcidieno, B., Giannini, F., Hassinger, S., Ovtcharova, J., (1994b),Feature based modelling by integrating design and recognition approaches, *Computer-Aided Design*, 26(8), pp. 646-653

Fields, M.C., Anderson, D.C., (1994) Fast feature extraction for machining applications, *Computer-Aided Design,* V. 26, No. 11, pp. 803-812

Henderson, Mark R. (1984), Extraction of Feature Information from Three Dimensional CAD Data, *Ph.D. Thesis,* Purdue University, West Lafayette, Indiana,

Kyprianou, L. K., (1980), Shape Classification in Computer-Aided Design, *Ph.D. Thesis*, University of Cambridge, Cambridge, England

Nau, D.S., Gupta, S.K., Kramer, T.R., Regli, W.C., Zhang, G., (1993) Development of Machining Alternatives based on MRSEVs, *Computers in Engineering*, ASME

Petta F., (1994), Design and development of a functional feature interpreter, (in Italian) *Thesis* aa. 1994/95 University of Genova, Italy

Pratt M.J., (1990) A hybrid feature-based modelling system, in *Advanced Geometric Modelling for Engineering Applications*, Eds. F.L.Krause, H. Jansen, Proc. GI-IFIP Conf. in Berlin, Germany, Nov.1989, North-Holland

Pratt M.J., (1993), Automated Feature Recognition and its role in Product modeling, *Computing* Suppl. 8, G.Farin, H.Hagen, H.Noltemeier (eds) Springer-Verlag, pp. 241-250

Sakurai, H., Gossard, D.C., (1990), Recognizing Shape Features in Solid Models, IEEE *Computer Graphics & Applications*, Vol. 10 No 5, pp. 22-32

Shah, J.J., (1991), Assessment of features technology, *Computer-Aided Design* Vol. 23, No. 5, pp. 331-343

Waco, D.L., Kim, Y.S. Kim, (1994), Geometric reasoning for machining features using convex decomposition, *Computer-Aided Design* vol. 26 No.6

Wilson, P.R., Pratt, M.J., (1988), A taxonomy of Features for Solid Modeling, *Geometric Modeling for CAD Applications*, Eds. M.J. Wozny, H.W. McLaughlin, J.L. Encarnaçao, Proc. IFIP WG5.2 Conf., Albany, NY, May 1986, North Holland

Vandenbrande, J.H., Requicha, A.A.G., (1990), Spatial Reasoning for Automatic Recognition of Interacting Form Features, *Computers in Engineering*, Eds. G.I. Kinzel, S.M. Rohde, D.W. Bennett, C. Born, A.A. Busnaina, G. Gabriele, V. Ghazarian, J.J. Shah, T. Shoup, Book n.G0515A, pp. 251-256