A Neutral Basis of Automatic Feature Recognition for Concurrent Engineering

Heng Zhu, Nien-Lung Lee* and Chia-Hsiang Menq

Coordinate Metrology and Measurement Laboratory (CMML)
Department of Mechanical Engineering
The Ohio State University
Columbus, OH 43210

Email: zhu@rclsgi.eng.ohio-state.edu, menq.1@osu.edu

Abstract

Multiple-way feature conversion/mapping has been identified as one of the key issues in feature technology that supports concurrent engineering (CE). In this paper, based on the homeomorphism concept and global Gauss-Bonnet theorem from differential geometry, a neutral basis that facilitates automatic feature recognition (AFR) is developed. Using the proposed neutral basis, basic features or neutral features that are domain independent can be extracted from boundary representation (B-Rep) models. The extracted basic features can be converted or mapped into domain-dependent features by utilizing domain-specific manufacturing information, thus they serve as the source for multiple-way feature conversion/mapping. The separation of general geometric reasoning about domain-independent features from specific reasoning about domain-dependent features provides a novel way to construct an information framework that tightly and efficiently integrates the neutral basis into the concurrent engineering environment. In the proposed neutral basis, bi-attribute is defined and abstracted from curvature computation and assigned to topological entities including faces, edges and vertices so as to group the entities having similar attributes and to form basic features. Feature Explorer, a proof-of-concept prototype system, has been implemented in C++ and ACIS on Windows NT platform to demonstrate the generality of the proposed approach and its ability to deal with real-world parts that are complex in geometry and have feature interactions.

1. Introduction

High competition in modern industry has brought more pressure for cutting costs, improving product quality and minimizing the time from concept to production. These requirements coupled with the ever-growing complexity of functional requirements of products and of production systems have contributed to a rising interest in life-cycle design optimization. The concept of concurrent engineering has been proposed to shorten the design cycle time and to increase productivity. Most of the design and manufacturing functions in concurrent engineering such as process planning, manufacturability evaluation, cost estimation and etc. require decision making using information suitable for reasoning about the important functionality in the application context. It is well recognized that low-level geometric and topological representation of a part is not sufficient to support the reasoning in modern design and production. Thus feature, a high-level semantic representation, was proposed to enhance the pure solid geometric models with useful engineering meanings, and it is believed that feature technology can bridge the gap between CAD and CAM to achieve design/manufacturing integration. Feature technology has been challenging the computer aided geometric design (CAGD) community as well as manufacturing community, and results in a major change of the functional role of a geometric modeler from being a drafting and modeling tool to being a common information model that supports product life-cycle.

Feature-based modeling can be utilized to create feature-based CAD models. Since the models are usually generated in specific context, they are not interchangeable among different applications. In addition, a large amount of legacy data from traditional solid modeling systems doesn’t have built-in feature information and it is very time-consuming to recreate those CAD models one by one in feature-based modeling. Therefore automatic feature recognition was proposed to extract features directly from pure solid models and different feature-based interpretation can be derived. Automatic feature recognition has become one of the emerging focuses of feature technology.

Current methods of automatic feature recognition often focus on the applications in machining such as process planning. The specific aim of these methods is to identify material removal volumes, or machining features. Thus it is
not really capable of addressing the issues in capturing the design intents. General-purpose automatic feature recognition should be able to extract any subset of geometric and topological entities from CAD models and needs to be flexible enough to support various manufacturing applications. Applications have their own domain knowledge and thus different interpretation of features. Therefore, it would be highly desirable to have a means to capture the higher-level neutral feature information that is domain-independent and convert the neutral features into domain-specific features that can be directly used by the manufacturing applications. The separation of general geometric reasoning about domain-independent features from specific reasoning about domain-dependent features can provide a novel way to construct an information framework that tightly and efficiently integrates the automatic feature recognition into the concurrent engineering environment.

In this paper, an overview of the information framework of the proposed automatic feature recognition for concurrent engineering is given in Section 2. Section 3 discusses the prior work on feature technology. Section 4 presents the neutral basis of the automatic feature recognition. Section 5 describes a proof-of-concept prototype system and an example is given. At last conclusions and future work are discussed in Section 6.

2. Overview of The Proposed AFR for Concurrent Engineering

An information framework that facilitates geometric reasoning in mechanical CAD/CAM functions for concurrent engineering is proposed below. Shown in figure 1, a neutral basis of automatic feature recognition is developed to bridge the gap between the geometric data of a design work and the higher level geometric abstraction that supports complex reasoning incurred in concurrent engineering environment. The geometric data of a design work can be originated from a traditional CAD modeling system, design by features, a sophisticated parametric design system, or from a reverse engineering process. The developed neutral basis can be applied to the different types of geometric data to extract basic features, including prismatic and rotational features, regular and irregular features. Depending on the specified downstream application, the basic features can be converted and transformed to the appropriate feature representations based on the rules from the domain knowledge databases of the applications such as automated tolerancing and inspection, automated process planning and etc. Thus the resulting higher-level geometric abstractions from the neutral basis can provide the source for multiple-way feature conversion/mapping to support concurrent engineering.

In the neutral basis of automatic feature recognition, the model geometry and topology are characterized and the invariant properties of various topological entities are identified. B-Rep model is adopted in this research work since it has the advantage of providing complete and unique object representation and clear separation between the geometric and topological descriptions of the object boundary. Bi-attribute is defined to classify three basic topological entities including faces, edges, and vertices. Based on these attributes, primitive forms and shapes of an object can be extracted and represented neutrally by basic feature categories. A feature processing kernel for concurrent engineering will be investigated and built in the future to transform basic features into the domain-dependent features. Multiple-way feature conversion and feature interaction resolving are the two key modules of this kernel. Application-specific information needs to be prepared from the manufacturing knowledge databases to help multiple-way feature conversion and feature interaction resolving. By using the feature

![Figure 1. Information Framework of Automatic Feature Recognition for Concurrent Engineering](image-url)
processing kernel, features for the downstream applications can then be recognized based on those basic feature categories along with attributed B-Rep information and manufacturing application-specific information. A prototype system has been implemented based on the geometric modeling engine ACIS [Sp98] to demonstrate the proposed neutral basis of automatic feature recognition.

3. Discussion of Prior Work

Features have been proposed as a means of providing high-level semantic information for interfacing CAD to manufacturing applications in product life-cycle [Pr93, Ma96]. Features are generally classified into the following categories [Sh91]: form features, precision features, material features, assembly features, and technological features. Among these feature categories, form features have been studied much more intensively than the other features, and from the geometric point of view, they can be further categorized as volumetric features that are solids and surface features that are collections of faces on a work-piece. Both include positive features and negative features that represent protrusions and depressions respectively. There have been three major approaches to create features [Sh91]:

- **Design by features or feature-based modeling:** This approach normally consists of several modules including feature modeling, feature library, validity check [Ch88, Sh90]. Parts are modeled by using pre-defined features from feature library. The drawback is that the feature model is not interchangeable among different applications.

- **Interactive or human-assisted feature recognition:** In this approach, a geometric model of a part is created first. Then, features are defined by the users through picking topological entities, associated with each feature, from a CAD model. The limitation of this approach is that it is time consuming and the recognized features could be user dependent.

- **Automatic feature recognition (AFR):** In this approach, form features of interest to an application are automatically extracted from the geometric model of a part. Thus designers are no longer restricted to the limited modeling method within predefined feature elements. Automatic feature recognition has fundamental significance from the integration perspective of product development cycle. A significant amount of research has been conducted for automatic feature recognition and can be classified into two major categories: graph-based approach [Ky80, He84, De87, Jo88, Fa89, Ma90, Ga90, Co91, Le95] and volume-based approach [Wo82, Pe90, Ki91, Va93]. In these studies, features are defined in terms of string characters, volumes, specific patterns or graphs consisting of faces, edges, and vertices. The recognition is normally performed by two procedures: first, characterizing faces, edges, or vertices; second, rules, grammar or graphs are applied to identify features by matching feature pattern in the database of the part model. More detailed survey on automatic feature recognition can be referred to [Ji97].

Most AFR approaches in the literature deal with a restricted set of features and are usually case-sensitive. This limitation is due to the lacking of systematic classification of the various geometric and topological entities in CAD models. Thus the identifiable features are normally simple in geometry. Moreover, if two or more features intersect, feature recognition using current approaches may become very difficult. In addition, recognizing features by dealing with low level topologic entities such as vertices, edges may suffer from combinatorial explosion problems [Ga92]. The variations in geometry and topology of features result in an overwhelmingly large number of patterns and create a substantial barrier to feature recognition. Most of current approaches are able to handle few sources of difficulty; however, practical problems often have many sources of difficulty occurring simultaneously. Thus, without a generic approach that can characterize various topological entities and their variations by using a set of common attributes, it would be difficult to recognize highly complex features that often exist in real world designs. Therefore, it is highly desirable to systematically characterize the geometry and topology and extract common attributes from low-level geometric data for identifying higher-level abstraction of a design that supports various life-cycle activities.

Recently multiple-way feature conversion/mapping have been identified as one of the key issues in feature technology to support concurrent engineering [Kr95, Br97]. In automatic feature recognition, multiple-way feature conversion/mapping is suggested to extract domain-specific features to facilitate the automation for needed applications. Most of the current automatic feature recognition methods focus on extracting machining features for process planning and cost evaluation. In order to facilitate its applications to other manufacturing processes, such as inspection planning, assembly/de-assembly and etc., more general and/or neutral features need to be considered. Neutral features are similar to the design features since both include positive and negative features. The advantage is that the extracted neutral features can capture the original design intents as much as possible from the pure solid models. Another merit of neutral features is that they can be converted or mapped into domain-dependent features by using the manufacturing knowledge bases, thus this new feature interpretation can efficiently bring AFR techniques into the concurrent engineering environment. For example, existing feature-based models do not consider technological information and functionality involved in manufacturing applications. Feature-based tolerancing can be used to augment the nominal geometric features with product tolerances and other non-shape technological attributes to achieve complete product models. The fully defined product models can be used to generate inspection features by mapping from design features and considering the non-shape technological information such as tolerance and datum. The identified inspection features
will greatly facilitate the automatic inspection planning of Coordinate Measuring Machines (CMMs) and tolerance evaluation. This consideration will give rise to the idea of integration of feature technology and CMM technology to achieve a new generation of CAD/CMM systems. In order to extract higher-level neutral features, a robust and general method based on solid theoretical foundation needs to be devised to create a functional module that performs geometric reasoning on CAD models.

4. A Neutral Basis of Automatic Feature Recognition

In order to develop the proposed neutral basis for automatic feature recognition, the necessary theoretical foundation and related issues are presented.

4.1 Theoretical Foundation

Part models with similar shapes may contain different geometric and topologic entities. For example, a cylindrical protrusion and a cylindrical depression are created on a rectangular block as shown in figure 2(a). If all the sharp edges of the protrusion and depression are replaced by fillets/rounds as shown in figure 2(b), the major global shapes of the protrusion and depression do not change but the geometric and topologic entities are different due to the local edge rounding. In order to recognize the major forms and local shapes from a part model, common attributes and invariant properties need be identified to characterize various geometric and topologic entities.

![Figure 2. (a) Cylindrical protrusion & depression on a block; (b) with edge fillets](image)

In this section several invariant surface properties derived from differential geometry and discrete topology as well as their applications to the derivation of the neutral basis of automatic feature recognition are investigated. The concepts of homeomorphism and global Gauss-Bonnet theorem [Ca76] are introduced and their applications to the derivation of bi-attributes, which characterize basic topologic entities, are developed as follows.

4.1.1 Homeomorphism

In topology, a topological space \((X,T)\) is defined as a set \(X\) for which a Topology \(T\) has been specified. The concept of homeomorphism is defined as follows.

**Definition:** Let \((X,T), (Y,S)\) be topological spaces and let \(h: X \rightarrow Y\) be bijective. Then \(h\) is a homeomorphism iff \(h\) is continuous and \(h^{-1}\) is continuous. If such a map exists, \((X,T)\) and \((Y,S)\) are called homeomorphic.

In other words, a homeomorphism, also known as topological invariance, is a continuous, one-one, and onto mapping between two topological spaces that has an inverse, and which is also continuous. For solid objects such as baseballs, donut, cups, etc., their shapes can be represented by the surfaces enclosing the objects. Ordinarily, such surfaces are closed, i.e. compact and without boundary. Two surfaces are considered to be homeomorphic if one of the surfaces can be continuously distorted to look like the other. Continuous distortion can be bending, stretching, and squashing without tearing or gluing points together. According to these criteria, the surfaces of baseballs, chalk, and bolts are homeomorphic. Similarly, the surfaces of nuts, the teacups with one handle, and the torus are homeomorphic also.

4.1.2 Global Gauss-Bonnet Theorem

For any compact orientable surface, a triangulation can be performed to divide the surface into a number of triangular patches. Let \(F, E, V\) denote the number of triangles, edges, and vertices, respectively on the surface after triangulation, the number

\[
\chi = F - E + V = 2 - 2g
\]

is called the Euler characteristic of the triangulation [Ca76], also known as the Euler-Poincare characteristic. In equation (1), \(g\) stands for genus, representing the number of handles added to a sphere. This shows that \(\chi\) is a topologic invariant of compact orientable surfaces in \(E^3\).

Given a triangulation of a compact orientable surface \(R\), and let \(C_1, \ldots, C_n\) be the closed, simple, piecewise regular curves which form the boundary of each triangle and \(\theta_1, \ldots, \theta_n\) be the set of external angles of the curves \(C_1, \ldots, C_n\), the global Gauss-Bonnet theorem can be expressed as

\[
\int_\partial R Kds + \sum_{i=1}^{n} \kappa_i(s)ds + \sum_{i=1}^{n} \theta_i = 2\pi \chi(R)
\]

where \(s\) denotes the arc length of \(C_i\), \(\kappa_i\) is the geodesic curvature of \(C_i\), \(K\) is the Gaussian curvature of \(R\), \(n\) is the number of piecewise regular curves, and \(p\) is the number of vertices after triangulation[Ca76]. The global Gauss-Bonnet theorem is important since it provides a relationship between the Euler characteristic that is defined in terms of topology and the total curvature that is defined in terms of differential geometry.

4.1.3 Form Features vs. Curvature

From the equation (1), we can know that a sphere and a rectangular block have the same genus, i.e.:

\[
g = 0
\]

because both of them are solids without handles. Therefore they have the identical Euler characteristic, i.e.:

\[
\chi = 2 - 2g = 2
\]
and the total curvatures are both $4\pi$’s since they are homeomorphic to each other. However, the distributions of the two curvature functions are very different. The Gaussian curvature of a sphere is a constant and is equal to the inverse square of the radius. On the other hand, the total curvature in the planar faces and straight edges of a rectangular block is zero and it is concentrated at the vertices.

If a protrusion is made on the top face of a rectangular block (as the cylindrical protrusion shown in figure 2(a)), the total curvature remains invariant since this deformation does not affect the Euler characteristic. It can be observed that even though the protrusion has positive total curvature, the intersecting edge between the protrusion and the top face of the block has concentrated negative total curvature that offsets the positive total curvature generated by the protrusion. If the sharp edge between the protrusion and the face is replaced by a smooth fillet as shown in figure 2(b), the total curvature still remains invariant. The negative total curvature to offset the positive total curvature of the protrusion is distributed on the fillet and can be calculated by $\int\int R\kappa d\sigma$. This example demonstrates that once a feature having positive total curvature is created on an object, there must exist a form that could be faces, edges, or vertices and has an opposite sign of total curvature to offset it.

From the above discussion, we can infer that adding one form feature to an existing object will lead to variations of curvature distributions. Curvature distributions can be characterized as feature properties and can be applied to automatic feature recognition. A further investigation in the classification of basic topological entities of B-Rep models for feature recognition is performed in Section 4.2.

4.1.4 Bi-attribute

Figure 3 shows three groups of two different features on the top face of a rectangular block. Applying the global Gauss-Bonnet theorem, we can find that all these features have positive total curvature and the boundaries intersecting with the planar face have negative total curvatures. The curvature distribution among the three terms in equation (2) for all the six features is listed in figure 4. From this figure, we can see that there is only one non-zero term in equation (2) for each feature in figure 3.

In other words, the non-zero total curvature can be obtained by faces such as features in group (a), edges such as features in group (b), or vertices such as features in group (c) in figure 3. However, features 1, 3, and 5, growing on the base surface, and the other features 2, 4, and 6, subtracted from the base surface, cannot be distinguished based on the sign of the total curvature. Therefore, additional criteria are needed to identify the differences between these two classes of features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$\int\int R\kappa d\sigma$</th>
<th>$\sum q\kappa_2(s)ds$</th>
<th>$\sum \theta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total surface curvature term</td>
<td>Total edge curvature term</td>
<td>Total vertex curvature term</td>
</tr>
<tr>
<td>1</td>
<td>$&gt;0$</td>
<td>$= 0$</td>
<td>$= 0$</td>
</tr>
<tr>
<td>2</td>
<td>$&gt;0$</td>
<td>$= 0$</td>
<td>$= 0$</td>
</tr>
<tr>
<td>3</td>
<td>$= 0$</td>
<td>$&gt;0$</td>
<td>$= 0$</td>
</tr>
<tr>
<td>4</td>
<td>$= 0$</td>
<td>$&gt;0$</td>
<td>$= 0$</td>
</tr>
<tr>
<td>5</td>
<td>$= 0$</td>
<td>$= 0$</td>
<td>$&gt;0$</td>
</tr>
<tr>
<td>6</td>
<td>$= 0$</td>
<td>$= 0$</td>
<td>$&gt;0$</td>
</tr>
</tbody>
</table>

Figure 4. Curvature distribution of the six features in figure 3

For those cases where the total curvature can be evaluated by surface curvatures, i.e., $\int\int R\kappa d\sigma$, the properties of Gaussian curvature $K$ can be applied to classify these surfaces. We know that Gaussian curvature is the multiplication of the two principal curvature $k_1$ and $k_2$:

$K = k_1k_2$

Also it is known that faces with positive Gaussian curvature ($K > 0$) can be classified into either concave or convex faces. So the sign of the two principal curvatures will determine the convexity of the face with $K > 0$. A convex face has two negative principal curvatures while a concave face has two positive principal curvatures. Faces with negative Gaussian curvature ($K < 0$) must contain mixed concave and convex properties since the two principal curvatures have different signs, i.e. one is positive and the other negative. These classifications result in three basic categories of form features: the positive feature referring to convex faces, the negative feature referring to concave faces, and the transition feature referring to mixed concave and convex faces, usually called transitive faces. We can find that transition features that consist of transitive faces always correspond to the fillets/rounds due to the local edge rounding.

According to the above criteria, feature 1 in figure 3 represents a positive feature because of convex face property while feature 2 represents a negative feature because of concave face property. The features 3, 4, 5, and 6 in figure 3 still cannot be evaluated by Gaussian curvature since the total curvature are concentrated at face discontinuities such as edges and vertices. Nevertheless, the concept of concave and convex properties can be extended to characterize edges and vertices. Thus, we employ the concept of concave and convex properties in describing a continuous curve to describe a curve containing curvature discontinuities. The “character edge” (defined in Section 4.2.2) is proposed to describe convexity discontinuities in a feature. The use of two attributes is similar to that of the two principal curvatures which are used to characterize a point on a surface. These two attributes will describe the two extreme curvature properties by using properties of concave, convex, and straight instead of numerical values like principal curva-
tures. Similar to the character edge, a vertex defined as the intersection of character edges can also have two attributes. The attributes of a vertex can be determined similar to a point on a character edge. Thereby, we introduce the definition of bi-attribute, \textit{BAtt} in short, to represent the two typical attributes of topological entities.

\textbf{Definition:} Let \( TE \) be a topological entity, then the Bi-attribute of \( TE \) is a two-tuple representation: \( \text{BAtt}(TE) = (b, b') \) where \( b \) and \( b' \) are the two typical attributes of topological entities.

For example, the two typical attributes of faces can be derived from the sign of their two principal curvatures. The detailed classification of faces, edges and vertices in bi-attributes is explained in the next section.

### 4.2 Classification of Topological Entities of B-Rep Models

In this section, three topological entities, namely faces, edges, and vertices, are considered since they are the basic constructive entities that form the B-Rep model of a part. Bi-attributes are used to classify these three topological entities as follows.

#### 4.2.1 Classification of Faces in Bi-attributes

As mentioned in Section 4.1.4, faces can be characterized by using signs of the two principal curvature \( k_1 \) and \( k_2 \). Thus, the proposed bi-attributes are specified by the signs of the two principal curvatures to characterize face \( F \) property as follows.

\[ \text{BAtt}(F) = (\text{sgn}(k_1), \text{sgn}(k_2)) \]

So the value set of the two components of the face’s bi-attributes is \([-1,0,1]\).

<table>
<thead>
<tr>
<th>Case</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( K )</th>
<th>( m )</th>
<th>Bi-attribute</th>
<th>Face Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;0</td>
<td>&lt;0</td>
<td>&gt;0</td>
<td>&lt;0</td>
<td>\text{BAtt}=(1,-1)</td>
<td>Convex</td>
</tr>
<tr>
<td>2</td>
<td>=0</td>
<td>&lt;0</td>
<td>=0</td>
<td>&gt;0</td>
<td>\text{BAtt}=(0,-1)</td>
<td>Concave</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>\text{BAtt}=(1,1)</td>
<td>Transitive</td>
</tr>
<tr>
<td>4</td>
<td>&gt;0</td>
<td>=0</td>
<td>=0</td>
<td>&gt;0</td>
<td>\text{BAtt}=(1,0)</td>
<td>Planar</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>&gt;0</td>
<td>&lt;0</td>
<td>&lt;0</td>
<td>=0</td>
<td>\text{BAtt}=(1,-1)</td>
<td>Transitive</td>
</tr>
<tr>
<td>6</td>
<td>=0</td>
<td>=0</td>
<td>=0</td>
<td>=0</td>
<td>\text{BAtt}=(0,0)</td>
<td>Planar</td>
</tr>
</tbody>
</table>

Figure 5. Classification of faces in Bi-attributes

If the principal curvature is less than zero, the attribute is \(-1\) and defined as convex. If the principal curvature is greater than zero, the attribute is \(1\) and defined as concave. Otherwise, the attribute is \(0\) and defined as straight when the principal curvature is zero. Mean curvature \( m \) is another important property of surfaces, and it is the average of the two principal curvatures. So it can also be used to classify faces by coupling with the Gaussian curvature \( K \). Hence by either using signs of the two principal curvatures \( k_1 \) and \( k_2 \) or using the signs of Gaussian curvature \( K \) and mean curvature \( m \), six cases of faces can be categorized into four types including convex faces, concave faces, transitive faces and planar faces as shown in figure 5. The planar faces do not belong to concave or convex categories. But it will be included in either positive or negative features as explained in Section 4.3.

#### 4.2.2 Classification of Edges in Bi-attributes

Edges with \( C^0 \) or \( C^1 \) continuities have curvature discontinuities and thus the total curvature cannot be evaluated by Gaussian curvature. However, from the global Gauss-Bonnet theorem in equation (2), the total curvature can still be evaluated by geodesic curvature along the edge. By applying the concept of homeomorphism, the geometric form of an edge can be altered to a surface such as a fillet and maintains the total curvature. This implies that we may also define bi-attributes to characterize the total curvature for an edge similar to the bi-attributes describing curvature properties of a point on a continuous surface. Thus, the definition of the bi-attributes of edges can be derived as follows.

From equation (2), the sign of total curvature from a curve can be determined by the sign of geodesic curvature \( k \). Since the geodesic curvature can be calculated from the curve and surface normal at the selected point, the sign of total curvature can then be determined. Similar to the classification of a face, the bi-attribute for an edge with negative total curvature will be concave and convex. For an edge with positive total curvature, the bi-attribute will be concave and concave or convex and convex which need to be determined by additional criteria. As mentioned earlier, positive total curvature at a selected point will have all concave or convex curves passing the point. Thus, any one curve can be used to identify the type of the bi-attributes for the selected point on an edge.

\[ \text{BAtt}(E) = (\text{sgn}(k_1), \text{sgn}(k_2)) \]

\( (-1: \text{convex attribute}; \ 0: \text{straight attribute}; \ 1: \text{concave attribute}) \)

\[ \text{Computation of the first attribute of edges’ bi-attribute} \]

Given an edge \( E \) and its two neighbor faces \( NF_1 \) and \( NF_2 \), and let \( NF \) be the edge’s owner face and \( NF \) be the edge’s partner face [Sp98], the retrieval operations of the edge’s owner face and partner face can be performed in ACIS as follows.

\[ \text{EDGE } E: \\
\text{FACE } \star NF1, \star NF2; \\
NF1 = ((\text{LOOP } *E->\text{coedge}()->\text{owner}())->\text{face}()); \\
NF2 = ((\text{LOOP } *E->\text{coedge}()->\text{partner}()->\text{owner}())->\text{face}()); \]

Let \( n_i \) and \( n_2 \) be the normals of the tangent planes \( TP_i \) and \( TP_2 \) of the neighbor faces \( NF_1 \) and \( NF_2 \); at a point \( P \) (usually take the parametrically middle point) on the edge, and let \( v \) be the tangent vector of the edge at the point along the edge direction (figure 6), then the first attribute \( b \) of the edge’s bi-attribute \( \text{BAtt}(E) \) is computed as follows.
shown in figure 7.

the practical machining and assembly. Smooth edges are very common in the models with fillets and rounds that are created in the modeling considering the practical machining and assembly. Smooth edges usually constitute part of the boundaries of fillets/rounds so they play an important role in recognizing these auxiliary features as one will see in Section 4.3. From the underlying geometry, smooth edges can be linear or non-linear.

The first attribute of all the smooth edges are set to 0 from the above computation. The computation of the second attribute of a smooth edge is different from that of a sharp edge because the first attribute of a smooth edge is zero and we can not decide the value of the second attribute just by using the geodesic curvature. By adopting the similar classification of smooth straight edges proposed by Kyprianou [Ky80], the second attribute of smooth linear and non-linear edges can be decided by the convexity of their two neighbor faces since the objects we consider are two-manifold objects.

The value set of the second attribute of smooth edges is extended to include 2 that indicates that one of the edge’s neighbor faces is transitive. The classification detail is shown in figure 8.

An example part is shown in figure 9 to illustrate all the above cases of smooth edges except case L6. The case L6 usually does not occur in the real-world parts because if a smooth linear edge has two planar neighbor faces, it will be merged into either of the neighbor faces and the two coplanar neighbor faces will be merged into one planar faces to form a regular two-manifold object.

<table>
<thead>
<tr>
<th>Case</th>
<th>NF₁</th>
<th>NF₂</th>
<th>Bi-attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Planar</td>
<td>Convex</td>
<td>BAₜ = (0,-1)</td>
</tr>
<tr>
<td>L2</td>
<td>Convex</td>
<td>Convex</td>
<td>BAₜ = (0,-1)</td>
</tr>
<tr>
<td>L3</td>
<td>Planar</td>
<td>Concave</td>
<td>BAₜ &amp;= (0,1)</td>
</tr>
<tr>
<td>L4</td>
<td>Concave</td>
<td>Concave</td>
<td>BAₜ &amp;= (0,1)</td>
</tr>
<tr>
<td>L5</td>
<td>Convex</td>
<td>Concave</td>
<td>BAₜ &amp;= (1,-1)</td>
</tr>
<tr>
<td>L6</td>
<td>Planar</td>
<td>Planar</td>
<td>BAₜ &amp;= (0,0)</td>
</tr>
<tr>
<td>NL1</td>
<td>Planar</td>
<td>Convex</td>
<td>BAₜ &amp;= (0,-1)</td>
</tr>
<tr>
<td>NL2</td>
<td>Convex</td>
<td>Convex</td>
<td>BAₜ &amp;= (0,-1)</td>
</tr>
<tr>
<td>NL3</td>
<td>Planar</td>
<td>Concave</td>
<td>BAₜ &amp;= (0,1)</td>
</tr>
<tr>
<td>NL4</td>
<td>Concave</td>
<td>Concave</td>
<td>BAₜ &amp;= (0,1)</td>
</tr>
<tr>
<td>NL5</td>
<td>Convex</td>
<td>Transitive</td>
<td>BAₜ &amp;= (0,2)</td>
</tr>
<tr>
<td>NL6</td>
<td>Planar</td>
<td>Transitive</td>
<td>BAₜ &amp;= (0,2)</td>
</tr>
<tr>
<td>NL7</td>
<td>Concave</td>
<td>Transitive</td>
<td>BAₜ &amp;= (0,2)</td>
</tr>
</tbody>
</table>

Note: NF means neighbor face.

Figure 8. Classification of smooth edges in bi-attributes

Classification of edges in bi-attribute

Whether the edges are smooth edges or sharp edges, generally they can be classified into three types: convex
edge, concave edge and character edge, and their definitions are as follows.

**Definition:** A **convex edge** is an edge with the bi-attribute of which two components are both $\leq 0$ and at least one is $-1$.

**Definition:** A **concave edge** is an edge with the bi-attribute of which two components are both $\geq 0$ and at least one is 1.

**Definition:** A **character edge** is an edge with the bi-attribute $(-1,1)$, $(1,-1)$ or $(0,2)$.

Character edges usually represent the convexity discontinuities in a B-Rep model. Character edges with bi-attribute $(-1,1)$ or $(1,-1)$ indicate the sharp discontinuities while the ones with $(0,2)$ represent transitive discontinuities such as the edges NL5, NL6 NL7 in figure 9.

### 4.2.3 Classification of Vertices in Bi-attributes

Similar to the edge, a vertex, which is defined as the intersection of edges, can also be classified by bi-attributes by applying the concept of homeomorphism. Ideally, bi-attributes of a vertex should be determined based on all the curves passing at the vertex on the adjacent surfaces. However, there is an easier way to determine the bi-attributes using the entire edges incident at the vertex in a B-Rep model. The bi-attribute of a vertex can be determined from the bi-attributes of the adjacent edges by using the operator similar to a logical “OR” operator. For example, if the entire edges incident at a vertex are concave ones, the bi-attribute of the vertex will both be concave as $(1,1)$ and the vertex is defined as a **concave vertex**. Similarly if the entire edges incident at a vertex are convex ones, the bi-attribute of the vertex will both be convex as $(-1,-1)$ and the vertex is defined as a **convex vertex**. If the edges incident at a vertex have both concave and convex ones, the vertex will have a convex and a concave attribute as $(-1,1)$. As long as there is at least one character edge existing among the entire edges incident at the vertex, the bi-attribute of the vertex is defined as $(-1,1)$ because the character edges represent the convexity discontinuities. In conclusion, the vertex with bi-attribute $(-1,1)$ is defined as a **character vertex** to represent the convexity discontinuity.

### 4.3 Recognition of Basic Features

The proposed bi-attributes can be used to characterize basic topological entities. Thus, the topological entities with similar attributes can be associated to form a higher-level geometric form. According to the definition of the bi-attributes, basic features such as positive features, negative features, transition features and fillets/rounds within positive/negative features can be extracted easily from the topological entities using the bi-attributes and they can be applied to extract higher-level feature information.

In addition to the bi-attributes, the proposed neutral basis comprises the positive and negative forms. The positive and negative forms provide not only the essential information for the forms and shape of a part model but also the links to feature extraction for various manufacturing processes. As discussed earlier, faces can be classified into four categories. Since edges and vertices can also be characterized by the bi-attributes, they are also classified into three categories. Thus positive forms can be identified by connecting convex topological entities, negative forms can be identified by connecting concave topological entities, and transition forms can be identified by connecting topological entities representing convexity discontinuities such as transitive faces, character edges, character vertices and their neighbor entities that surround positive or negative forms.

#### 4.3.1 Characterization and Identification of Feature Loops

The feature loop is defined as the connected topological entities with common attributes. Despite its name, a feature loop is not restricted to comprising only edges and vertices. It may comprise adjacent entities of vertices, edges, or faces. A single face can be defined as a feature loop, for example a simple through hole consists of only one cylindrical face. A depth-first searching technique is applied to identify forms by searching the adjacent topological elements. Two categories of feature loops can be identified as follows.

- **Positive feature loop** (PFL): Starting with any face, edge, or vertex with bi-attribute $(-1,-1)$, connect the adjacent face, edge, or vertex with bi-attributes including $(-1,-1)$, $(-1,0)$, $(0,-1)$, or $(0,0)$.

- **Negative feature loop** (NFL): Starting with any face, edge, or vertex with bi-attribute $(1,1)$, connect the adjacent face, edge, or vertex with bi-attributes including $(1,1)$, $(1,0)$, $(0,1)$, or $(0,0)$.

The adjacency relationship is defined based on the connectivity among basic topological elements in B-Rep models. By applying the searching technique, the positive and negative feature loops can be identified. Note that vertices and edges with $(0,0)$ attributes and planar faces can be included in both types of forms.

#### 4.3.2 Characterization and Identification of Character Loops

After performing identification of feature loops, character loops can then be identified. A character loop can be simply defined as the adjacent faces, edges, or vertices with negative or zero total curvature surrounding a feature with positive total curvature. Since there are two types of features, two categories of character loops can be identified as follows.

- **Positive character loop** (PCL): Starting with any face, edge, or vertex with bi-attribute $(-1,1)$, $(1,1)$ or $(0,2)$ which is adjacent to a positive feature, connect the adjacent face, edge, or vertex with bi-attributes including $(-1,1)$, $(1,1)$, $(0,2)$, $(1,0)$ and $(0,1)$ to surround the positive feature.

- **Negative character loop** (NCL): Starting with any face, edge, or vertex with bi-attribute $(-1,1)$, $(1,1)$ or
the adjacent face, edge, or vertex with attributes including \((-1,1), (1,-1), (0,2), (0,-1)\) and \((-1,0)\) to surround the negative feature.

Note that a character loop normally comprises a closed chain of basic topological elements for simple isolated features. However, a character loop need not be closed, especially when features interact with each other.

### 4.3.3 Construction of Entity-Loop-Attribute Graph

After performing identification and characterization of faces, edges, vertices, feature loops, and character loops, a complete description of adjacency relationships among these elements and their properties can be illustrated by a graph, namely an Entity-Loop-Attribute (ELA) graph. An ELA graph is the abstraction of the data structure that contains feature loops, character loops, vertices, edges, faces, and their attributes in addition to the B-Rep data structure.

![Figure 10. An example part and its primitive topological entities](image)

Figure 10. An example part and its primitive topological entities

An example part and its primitive topological entities are illustrated in figure 10. By applying the comprehensive classification of topological entities, we can characterize all the faces, edges and vertices in bi-attributes as follows.

**Faces:** (0,0): \(F1,F2,F3,F4,F5,F6,F7,F10,F15,F16,F17,F18\)

\((-1,0): F8,F11,F12,F13,F14\)

\((-1,1): F9\)

**Edges:** \((-1,0): E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12\)

\((0,-1): E13,E14,E15,E16,E17,E18,E19,E20\)

\((1,0): E21,E22,E23,E24,E29,E30,E31,E32\)

\((1,-1): E25,E26,E27,E28\)

\((-1,-1): E33\)

\((0,2): E34,E35\)

**Vertices:** \((-1,-1): V1,V2,V3,V4,V5,V6,V7,V8\)

\((-1,1): V9,V10,V11,V12,V13,V14,V15,V16\)

\((1,1): V17,V18,V19,V20\)

After the classification of topological entities, feature loops and character loops are identified and represented in an ELA graph as shown in figure 11.

In an ELA graph, an elliptic element represents a feature loop or a character loop. The type of each topological entity is included in the B-Rep data structure and can be referenced from the entity's ID. For a feature loop, it may include the elements with the same type of attribute and the pointer of the character loops, if any, enclosed by the feature loop. For a character loop, it usually contains the pointer of the enclosed feature loop.

![Figure 11. ELA Graph of the part shown in figure 10](image)

Figure 11. ELA Graph of the part shown in figure 10

An arrow that is only used between two loops represents the parent-child relationship between two loops. Two loops are considered to have parent-child relationship when one of the two loops is completely enclosed by the other where the child loop is enclosed by the parent loop. A child feature is normally composed of two loops: one character loop and one feature loop. The connections among loops in the ELA graph are similar to a tree structure. The base feature is the root node in the tree structure and it contains several leaf nodes that correspond to the features added on the base feature. The processes of adding features on other features are equivalent to those of adding new leaves on the existing nodes. Each terminate node may represent a feature or part of a feature. Therefore the recognition of basic features can be performed by tracing the tree structure.

### 4.3.4 Classification and Recognition of Basic Features

Basic features are represented in terms of faces, edges, vertices, feature loops, and character loops. Four categories of basic features are defined as follows.
• **Base Feature**: A base feature is defined as a basic form or volume to start with in design or manufacturing. It normally consists of a simple geometry, such as a rectangular or a cylindrical stock. A base feature usually corresponds to the root node of the tree data structure in the entity-loop-attribute graph.

• **Positive Feature**: A positive feature is defined as a form with convex property added on a part where the intersecting boundaries between the feature and the part have negative total curvature property. The convex form is defined as the adjacent vertices, edges, and faces having convex-and-convex or convex-and-straight attributes. A positive feature containing planar faces is allowed. In the ELA graph, a positive feature usually corresponds to a positive feature loop.

• **Negative Feature**: Contrary to the positive feature, a negative feature is defined as a concave form by removing a form with convex property from a part, where the intersecting boundaries between the feature and the part have negative total curvature property. Similar to positive features, planar faces are also allowed in negative features. In the ELA graph, a negative feature usually corresponds to a negative feature loop.

• **Transition Feature**: A transition feature is defined as a form representing segmentation among features. It can comprise a connected portion of vertices, edges, or faces with negative total curvature. Thus, transition features will contain both concave and convex properties. Note that edges or faces with zero total curvature may be identified as transition features when they are adjacent to a feature loop where the non-straight attribute is different from the attribute of the feature loop. In the ELA graph, a transition feature corresponds to a positive or negative character loop. Transition features with faces form fillets/rounds between different positive and negative features.

• **Fillets/Rounds within Positive/Negative Features**: Transition features only represent a subset of all the possible fillet/round features in the part. Fillets/rounds often exist in positive/negative features due to the functional necessity or machining process. From the design view, fillets/rounds are formed by blending the sharp edges and vertices. The blending operations will cause the generation of smooth edges. Therefore, in order to capture these important detail design intents, smooth edges are used to provide the hints to recognize the fillets/rounds after the recognition of positive/negative features.

**4.3.5 Refinement of Recognized Features**

Due to the complex feature interactions, feature loops and character loops are broken into several groups that form separate positive and negative features. Actually these positive/negative features should be represented by one positive/negative feature. So feature refinement needs to be performed to optimize the recognition result. Two steps are executed to achieve this goal: (1) concatenate all the positive features that are adjacent, i.e. not separated by the character loops; (2) concatenate all the negative features that are adjacent, i.e. not separated by the character loops.

After feature refinement, complete discrete positive and negative features can be distinguished by the positive or negative feature loop of the feature. Transition features with faces and fillets/rounds within positive/negative features form the whole set of all the auxiliary features in the B-Rep models. In order to recognize domain-dependent features for manufacturing applications, positive features, negative features, and auxiliary features can be reorganized and classified into sub-categories according to the distinct characteristics and behaviors of downstream manufacturing applications.

**5. Feature Explorer – A Proof-of-Concept Prototype System**

The conceptual framework of the proposed neutral basis for automatic feature recognition has been implemented on Windows NT platform by using Visual C++ and ACIS, a current leading geometric modeling engine in object-oriented structure [Sp98]. The proof-of-concept prototype software system, called Feature Explorer, consists of five functional modules (figure 12) including Geometric and topological preparation, Classification of topological entities (faces, edge and vertices) in bi-attributes, Identification & characterization of feature loops and character loops, Construction of Entity-Loop-Attribute (ELA) graph, Recognition of basic features. The geometric and topological preparation module creates the adjacent entity lists for all the primitive topological elements including faces, edges and vertices. The generated entity adjacency information is used in the loop identification module to reduce the effort of retrieving adjacent entities.

A Graphical User Interface (GUI) (figure 13) has been designed for the prototype system to allow the users to visualize the feature recognition processes, interrogate the basic
features in the B-Rep models, and investigate the bi-
attributes of all the topological entities including faces,
edges and vertices that the features consist of. As we can see
from figure 13, negative feature No. 17 is currently investi-
gated. After mouse picking on the feature, it is highlighted
and a dialog is popped up to let the user know its feature
type and feature ID. With such information in mind, the
user can activate the basic feature browser to list all the
found features and expand the subtree under the interested
feature to check its component faces, edges and vertices and
their bi-attributes.

Many real-world parts have been tested in the prototype
system and the results show that it can robustly extract dis-
crete basic features and handle nested feature interaction
without difficulty. An example is given below to demonstratethe
generality of the proposed approach.

The hidden-line removal model in figure 14 shows a
typical part with basic features created on a rectangular
block. Discrete basic features and some interactions exist in
the part. All the recognized basic features are illustrated in
figure 15. Those basic features represent all the categories
of basic features and contain cylindrical, toroidal and planar
surfaces. In figure 15, positive features including the base
feature are represented in cyan color. There are nine posi-
tive features: one base block, two islands (cylindrical and
rectangular), two bridges (cylindrical and irregular), one
rectangular block, three bosses (rectangular, cylindrical and
irregular). Negative features are shown in yellow color and
twenty-six discrete negative features are recognized from
the example part as indicated in figure 13. Transition fea-
tures that separate features are represented in magenta
color. Fillets/rounds formed from those transition features
that have face entities. In addition, fillets/rounds are also
identified within the positive and negative features and in-
dicated in green color to show the difference from the fil-
lets/rounds generated from transition features. It is worthy
noting that as long as the feature interactions do not change
the property of feature loops and character loops remain
unsplit, nested feature interactions and simple positional
feature interactions can be handled without difficulties as
shown in figure 13. Nested feature interactions usually refer
to the typical parent/child feature relationship such as the
two recognized islands located within the irregular pocket.
feature. Positional feature interactions result from the fact that at least two features interact with each other because of their relative positions. At the irregular corner step, positional interactions among several negative features occur. Since character loops remain unbroken, four negative features (irregular pocket, irregular passage, rectangular passage and irregular corner step) are successfully identified without further reasoning. The same situation can be found in the cylindrical boss with several negative features such as through slot and rectangular passage.

6. Conclusions and Future Work

This paper presented a neutral basis of automatic feature recognition for concurrent engineering. In this approach, the concept of homeomorphism and global Gauss-Bonnet theorem are utilized and bi-attribute is defined and abstracted from geometric and topological computation. Based on this theoretical foundation, a comprehensive classification of topological entities (faces, edges and vertices) in bi-attributes is performed. The attributed entities are grouped according to their attributes to form feature loops and character loops that are used to construct ELA graphs. Finally the proposed basic features are extracted from ELA graphs. The proposed neutral basis can be applied to a wide variety of geometric models and feature categories as demonstrated by the implemented prototype system. In order to fully integrate the proposed neutral basis into a concurrent engineering environment, two key issues will be considered in the future work: (1) based on a comprehensive classification of various feature interactions, robust feature interaction resolving methods need to be devised; (2) based on the characteristics of downstream manufacturing applications, flexible multiple-way feature conversion/mapping mechanisms will be constructed to convert the basic features into application-specific features.

7. Acknowledgement

The research is being supported by the National Science Foundation through grant numbers DMI-9713768. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. References