Volumetric Feature Recognition for Direct Engineering

Edward J. Bezdek Richard H. Crawford Kristin L. Wood

David Thompson

Notation

FA	Feature
VA	Volume
fA	Face
e	An edge
vA	Vertex
G, H	Graphs
TA	Topological space
XA	General set
$b(X)$	The boundary of set X
$i(X)$	The interior of set X
$c(X)$	The closure of set X
x	An element of a point set
S	A system of sets, or family of sets
$Pred_P(X)$	A predicate or open statement; either true or false
$\{X \mid P(X)\}$	The set of all X such that $P(X)$ is true
$Rel_r(x,y)$	A relation containing the ordered pair (x,y)
$\wedge$	Logical “and”
$\vee$	Logical “or”
$\neg$	Logical negation
$\cap$	Boolean set intersection operator
$\cup$	Boolean set union operator
$\cap^*$	Regularized Boolean set intersection operator

$\cup^*$	Regularized Boolean set union operator
$-^*$	Regularized Boolean set difference operator
$\oplus$	A set operator defined such that $(A \oplus B) = (A \cup B) - (A \cap B)$
$\bigcup_{i=1}^{n} c_i$	The distributed union = $c_1 \cup c_2 \cup \cdots \cup c_n$
$\subset$	Is a proper subset of
$\subseteq$	Is a subset of
$\supset$	Is a proper superset of
$\supseteq$	Is a superset of
$\in$	Is an element of
$\overset{\max}{\in}$	Is a maximal element of
$\notin$	Is not an element of
$\forall$	Universal qualifier denoting ``for all"
$\exists$	Existential qualifier denoting ``there exists"
$\Rightarrow$	Implication
$\Leftrightarrow$	If and only if
FBD	Feature Based Design
BV	Base Volume
MSF	Maximal Simple Feature
VFR	Volumetric Feature Recognition (method)
IGL	Intermediate Geometry Language
EVFR	Extended Volumetric Feature Recognition (method)

1. A Study in Contrasts

A feature provides a shorthand by which information can be communicated quickly and efficiently. At their essence, features are *an efficient and powerful means of information transfer*. To engineers, “features,” and the information they represent, are of interest in a variety of types of analysis. The particular type of analysis defines what specific “features” are germane. In an engineering environment, features are often associated with both manufacturing and design information.

It is more or less agreed upon amongst researchers that a feature is a “physical part of an object being mappable to a generic shape and having functional significance” [49]. Beyond this, attempts at defining “generic shape” and “functional significance” often involve generating large taxonomies of feature shapes and functions (see, for example, Shah [44]).

Research on direct engineering systems using features is promising, though fully mature systems have yet to appear. Krause, et al. [31] make use of “semantically endowed objects” (i.e., features) to aid in integrating the various steps of the product development environment. Fu and Nee [18] explore the problem of converting between various feature viewpoints in support of a concurrent (or direct) engineering environment. Xue and Dong [53] formalize sets of design and manufacturing features, organized via a fuzzy c-means clustering algorithm developed by Bezdek [2], and use them to develop a prototype system that automates a large portion of the design cycle.

In most -- if not all -- modern concurrent and direct engineering systems, the common design database is manifested by a solid modeler of some sort. The geometric representation of the part serves as the common basis on which the functions of the various steps in the design process operate. For example, the designer creates a conceptual model of the part in the geometric modeler and it is immediately available to downstream manufacturing engineers who can begin to analyze it for potential difficulties. When an application from a particular domain needs to make use of a feature stored in the database, it extracts the particular type of “form feature” of interest and attaches to it the necessary application-specific information (i.e., semantics).

The primary advantage of a feature-based design (FBD) philosophy is that the high level feature information is not lost in a strict, geometry-based database representation. Many researchers consider the question “why should we discard the feature-level information only to have to reconstruct it whenever a database request is made?” and decide that, indeed, they need not. Of course, then the next question that must be answered is “what features will our design be based on?” One alternative is to allow the use of *every* different type of feature from design through manfacturing to assembly. This, of course, quickly creates an unmanageable situation -- both in general and more specificially with any existing direct engineering database. Using all available types of features results in a database where the large majority of the information is worthless to any given user.

As a solution to this problem, some feature-based design schemes simply choose a single set of features and mandate that it be used in the design process. Most commonly, this chosen feature set is a set of machining features, or at least manufacturing features. Designers are required to build their part designs with this manufacturing feature set, which is in turn processed by downstream CAM applications. Of course, this assumes that the designers have knowledge of all the manufacturing processes available and, more importantly, that designers may not store *design* information effectively. Thus, some technique is needed that can extract feature information appropriate to the task at hand.

Some researchers have instead tried to develop a means by which design features can be mapped or translated into manufacturing features [41]. This allows the designer to work in terms of design features (ostensibly allowing for an unencumbered creative process), which are then mapped into the appropriate type of manufacturing features for CAM applications. Other researchers have attempted to create a unified feature set from which both design and manufacturing features can be derived [22, 4, 6]. However, these approaches are founded on the theory that all necessary feature information can be extracted in this fashion. In fact, the features contained in a feature-based design representation can in fact combine in such a way so as to create a valid feature that is *not* part of the scheme. The first case study presented in Section 3 shows such an example. So even *if* an ideal feature set could be arrived at with which to build designs, a feature recognition technique would still be necessary to consider all appropriate feature interpretations.

But perhaps the most pressing reason for feature recognition is the question “why should designs be `based' on features at all?” FBD techniques work well with modern geometric modelers because such modelers are parametric in nature (i.e., they use parameters to define features, which are in turn used to define parts). But why do modelers have to be parametrically based? More advanced geometric modelers are starting to use a "free-form" sculpture paradigm wherein the designer virtually sculpts a part to create the final design. In such an environment, the designer is not thinking at all in terms of features, yet they still might prove useful in downstream applications. In order to be used in these downstream applications, a feature recognition method will once again need to be employed.

1.1 The problem of feature recognition

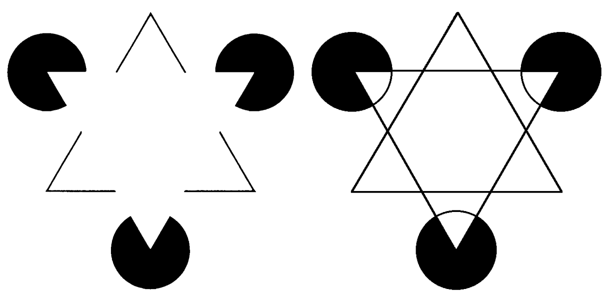
Conceptually, feature recognition is simply a process of inference. To infer something, one makes a judgement based on a set of incomplete data. This section describes four tasks which are difficult for computers to accomplish and thus make the task of feature recognition a daunting one.

1.1.1 Converting low level to high level

When beginning with a low-level model representation such as a BRep, a recognition scheme must translate the low-level representation of faces, edges, and vertices into a high level representation of form features. A similar translation is required to move from form features to semantically-endowed features. Features are useful when they communicate a high level of semantic meaning. When parts are stored in a BRep fashion, the specific geometric configuration that makes up a form feature is lost and must be reconstructed in order to be used again. In short, the scheme must infer the presence of the higher level feature entities from the lower-level geometric and topological entities.

1.1.2 Dealing with incomplete data sets

The previous section described one type of inference that eludes computers -- that of inferring from one level of abstraction to another. But another type of inference is equally difficult for computers to make, namely, inferring the complete nature of an incomplete data set. Consider Figure 1(a) as an example.



(a) Is there a triangle? (b) Patterns restored.

Figure 1 The problems of feature interactions and incomplete representations.

1.1.3 Considering multiple interpretations

Beyond even the problem of incomplete data sets lies another issue with which computers deal poorly, namely, multiple interpretations. In other words, given a complete set of data, a computer will typically arrive at one conclusion.

1.1.4 Of "grounds" and "figures."

Finally, we consider what is perhaps the most difficult challenge that faces a feature recognizing computer. Allowing for the moment that the machine is able to overcome the first three problems described, consider the inferrence that must be made to appreciate the significance of not only what data is *present* in the set, but what data is *absent* from it. Much in the way that the rests are as important as the notes in a Beethoven symphony, sometimes there is meaning not only in what geometry is present in the data structure, but also what geometry is *not* present.

1.2 Current feature recognition methods

The plethora of methods developed in the research community is made up of techniques as varied as they are numerous. A complete survey of all feature recognition techniques would be a serious undertaking and is beyond the scope of this work. A numer of excellent comprehensive reviews of feature technology exist, including [45, 47].

The techniques described herein will focus on five major areas of research. The first is the so called "graph matching" method.

The method is well characterized by the work of Joshi and Chang [29] and their attributed adjacency graph (AAG), which is a graph based on concavity or convexity with respect to the part volume. The recognition process then proceeds by searching for feature graphs within the larger part graph.

Other researchers active in the field include Chuang, et al. [9, 8], Fu, et al. [17], Horaud, et al. [27] and others [38, 10, 34, 41].

Comparing graph matching approaches to the four criteria developed in Section 1.1, it is apparent that this type of method adequately infers high level entities from low-level entities (though each high level entity must be explicitly defined), but the other three points are not well met. Obscured representations are difficult to recognize; typically only one "feature interpretation" of a part is generated; and the method does not consider the "ground" that might be suggested by the pattern's "figure." Attempts have been made to address some of these shortcomings. For example, the problem of patterns being obscured in the larger part representation has been tackled by Ji, et al. [28] and their "virtual links." Another intriguing attempt to solve this issue is the so called "hint processing" technique.

A second method for feature extraction is feature "hints," originally developed by Vandenbrande and Requicha [50, 25, 26]. The input (i.e., a geometric model) is processed by production rules that generate hints for the presence of machining features. An example of a geometric hint is two parallel faces which suggest the presence of a SLOT feature. In this respect, a hint processing technique essentially searches for partial feature graphs, avoiding the problem of canonical feature graphs being obscured by feature interactions. Hints are also acquired from more information than just edge graphs, thus using more of the information available.

Referring again to our four requirements, the hint processing method addresses most of the points. The main problem is that, in order to recognize a general feature from a general geometry, *every* possible feature (a potentially infinite number) must be specified beforehand.

Pre-defining such a large number of features is a difficult -- if not impossible -- task. The issue is made worse by the fact that as each group of researchers develops a recognition method, they develop their own set of pre-defined features. In other words, no standard exists. Gupta, Regli, Nau, et al. [40, 24, 39] have attempted to address this issue by basing their recognition method on a standardized set of pre-defined features, namely the material removal shape element volumes (MRSEVs) as defined by Kramer [30]. The drawbacks of the method include the fact that it is defined solely in terms of machining features, and so by definition cannot recognize the "ground" suggested by the "figure" of a machining feature.

Yet another technique developed by Gadh and Prinz [21, 19, 20] defines features with loops of edges ("c-loops") from the part that are either concave or convex to the part volume. One advantage of the search method is that it's independent of the context (i.e., design, manufacturing, etc.)

However, there are problems handling interacting feature representations and in generating multiple feature interpretations.

A technique which circumvents obscured edge and face representations would solve many of the problems with the methods reviewed above. Volume decomposition approaches achieve this by reconstructing this missing topology by breaking down the part volume into smaller sub-volumes and then reconstructing the sub-volumes in features.

One of the earliest forms of volume decomposition was proposed by Woo [52] and was known as the Alternating Sum of Volumes (ASV) method. The ASV method successively takes the difference of a part from its convex hull, and then takes the difference of the difference from *its* convex hull, etc. until the difference operation returns the null set. The original ASV method had a serious problem in that it would not converge upon a solution in certain degenerate cases. This problem, as well as the original method's restriction to prismatic volumes, is solved in subsequent work by Kim, et al. [51, 33]. Furthermore, their method -- christened Alternating Sum of Volumes with Partitioning (ASVP) -- has been further extended [37] so that when changes occur to the model only localized updates with the ASVP method need occur rather than global updates. Like the incremental technique of Han and Requicha [25], the localized updates make the ASVP well-suited to a direct engineering environment.

Another volume decomposition technique involves surface extension to reconstruct the missing topological pattern of a feature. As an example of surface extension, consider again Figure 1(a). Upon closer examination of the figure, one notices that in fact all the requisite edges exist to define the triangles and circles suggested in the picture, but that they are obscured or cut-off at various points in the image. By *extending* the edges of the image, the image in Figure 1(b) is arrived at.

After the part faces are extended, they are used to separate the original part volume into smaller “cells” or “base volumes.” These cells are subsequently reconstructed into “maximal” or “maximal simple” features. The semantic labels differ from researcher to researcher, but the various techniques are quite similar.

This technique is well-characterized by the work of Tseng and Joshi [48], Shen and Shah [46], and Sakurai [42], but initially was limited to prismatic parts with planar faces. Recent work by Sakurai and Dave [43, 16] and Coles, Crawford and Wood [12] has extended the technique to apply to more generally defined parts (i.e., those with quadratic surfaces).

All of the volume decomposition methods suffer from a problem of combinatorial complexity. The process of taking a single volume, decomposing it into numerous smaller volumes, and then unioning those smaller volumes back into larger feature volumes is an involved and complicated process. Furthermore, when the feature volumes recognized are combined in an exhaustive fashion to construct multiple feature interpretations of a part, the CPU effort required is significant, and many of the interpretations found are non-intuitive.

However, the volume decomposition method performs well when judged by the four criteria of Section 1.1. The only point it fails to measure up against is the fourth. The method, as defined, does not simultaneously consider the “figure” and the “ground” interpretations of the part geometry. Coles, et al. [12], however, have defined their method such that it will recognize both subtractive *and* additive features, though it does not do so simultaneously.

One final recognition technique deserves mention. The work of Lee and Menq [32] is unique from those methods described previously in that it defines form features solely in terms of the *curvature distribution* across the surface of a given part.

The technique only meets two of the criteria specified in Section 1.1. As specified in the referenced paper, it does not generate multiple feature interpretations, and it cannot recognize as separate two intersecting features.

However, what is unique and exciting about the method is that it deals solely with the surface curvature of the part. What this means is that it could be used in conjunction with a “free form” computer-aided design tool wherein the designer virtually “sculpts” a part. This sort of tool is not presently available, but represents a more advanced form of modeling tool that will appear in the future.

1.3 Summary of unresolved issues

Given then that feature recognition is crucial to the success of feature technology, research aimed at improving recognition techniques is of great benefit to the field of computer-aided engineering. One item to note is that the majority of the methods presented earlier in this chapter are unable to recognize both subtractive and additive features using the same methodology. What's more, even if they can consider both subtractive and additive features (perhaps using similar methodologies), the two types of features are rarely recognized as part of a single integrated theory. This prevents them from being considered simultaneously as part of a unified analysis using the results of the recognition method.

2. Foundation and Definitions of the Theory

This section outlines a theory of feature recognition method based upon two separate yet complementary theories previously developed at The University of Texas at Austin, namely, a methodology for feature recognition (the Volumetric Feature Recognition or “VFR” method), and a formalism for representing interactions and relationships between features (the Intermediate Geometry Language, or “IGL”). Represented herein is an übertheory that integrates the two while expanding upon them to increase their scope and address some of their recognized shortcomings. For a thorough explanation of the original theories, the reader is referred to the seminal works, specifically, of Coles, et al. [11] regarding the VFR and of da Silva [14] and Navaneethakrishnan, et al. [35] concerning the IGL.

At its essence, feature recognition links the base geometry of a part — surfaces, curves, and points — directly to specific features, and thus to manufacturing processes and ultimately to cost. By knowing the portions of the geometry of the part volume that is associated with specific features and thus with specific portions of the manufacturing cost, later revisions to the design can be made with this information in mind. In other words, variant design and direct engineering can be performed with respect to manufacturing cost information.

A feature recognition method generates one or more “descriptions” of a part, using form features as its language.[1] This feature description may describe how the part is designed, or how the part is manufactured. For the purposes of this work, we focus primarily on manufacturing analysis.

The manufacture of a part may be described in terms of a stock volume V_S and a part volume V_P.[2] More specifically, a part volume may be expressed in terms of the stock volume and the manufacturing processes necessary to create V_P from V_S. Form features provide a means to represent manufacturing processes, so a part volume can be described by a stock volume and a set of form features. Symbolically, $V_P = V_S \bigcup_{i=1}^{n} F_{a_i} \bigcup_{j=1}^{m} F_{s_j}$, where F_a is an additive form feature and F_s is a subtractive form feature.

---

[1] The concept of form features as a “language” is discussed at length in [5].

[2] All manufacturing processes can be thought of as beginning with a stock volume. This is true even for those methods that deal solely with additive manufacturing processes. In such cases, the stock volume, as it were, is empty (i.e., $V_S = \varnothing$).

Depending on the stock volume chosen, differing sets of form features are necessary to describe the part volume. Most of the feature recognition techniques discussed in Section 1.2 are limited to generating feature sets of a single type (either additive or subtractive). Those that do deal with both additive and subtractive features either do so via slightly different methods, or else they implicitly define a "base feature" that corresponds to the stock volume V_S. The theory presented herein (hereafter referred to as the Volumetric Feature Recognition or "VFR" theory[1]) generates a set of features of potentially mixed types from an arbitraryily defined stock volume. Furthermore, it does so via a single definition for both additive and subtractive features. In this way, it is unique among the methods of Section 1.2.

The sections that follow present the theoretical definitions that define the VFR theory. The theory makes use of the concept of surface extension, and in that respect it is similar to other surface extension based methods discussed in Section 1.2. However, in defining additive and subtractive features using a single definition, by recognizing both types of features simultaneously, and in generating mixed feature sets from an arbitrarily defined stock volume, it extends the functionality of the "typical" surface extension based recognition method.

2.1 Basic concepts

This section presents some of the basic concepts necessary to provide a foundation for the more complex entities defined below. This section and those that follow use topological and geometric concepts that are rigorously defined in [11]. Figure 2 displays an example part that will be used throughout the section to illustrate the theory.

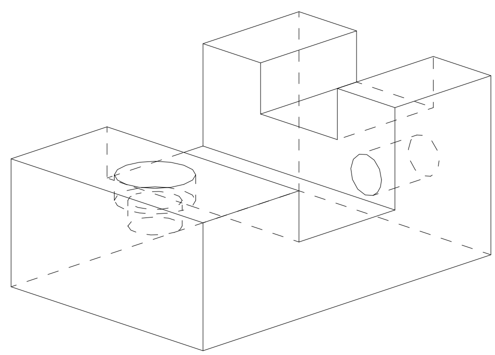


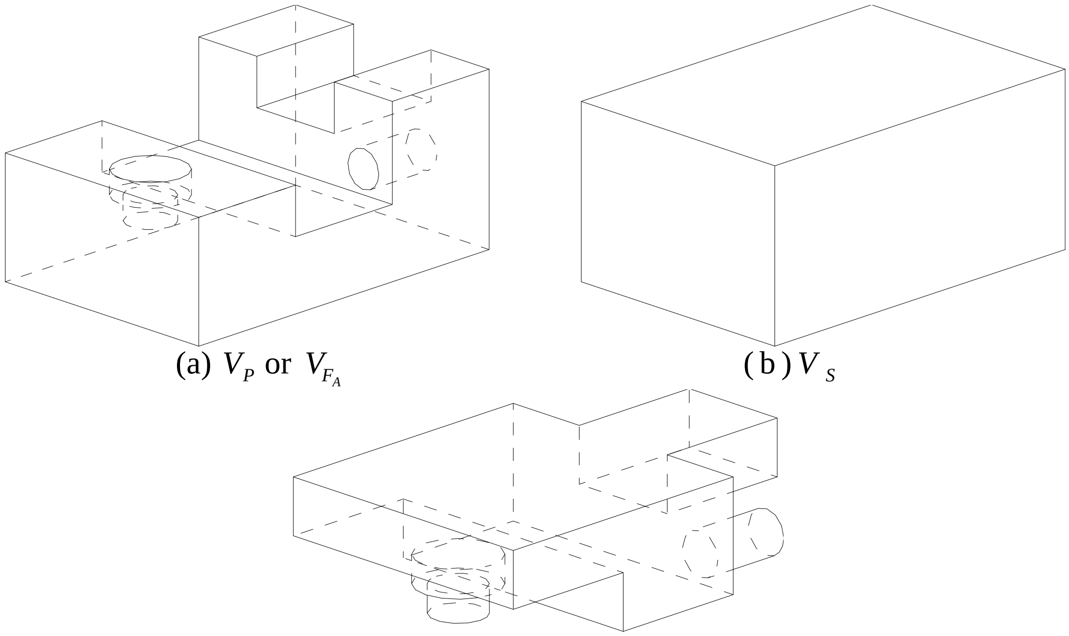
Figure 2 An example part

2.1.1 Types of volumes

A "volume" is a closed, connected, regular subset of a three-dimensional space. Several types of volumes are of use in the VFR theory. First, there is the volume of the part V_P (Figure 3(a)) from which features are recognized. Second, there is the volume of the stock V_S (Figure 3(b)) from which the part is manufactured. In addition to V_P and V_S, most feature recognition theories define a "delta" volume V_Δ (Figure 3(c)) equal to the boolean difference between the stock volume and the part volume. Written symbolically, $V_\Delta = V_S -^* V_P$. The delta volume coupled with the part volume form the foundation of most feature recognition theories. They are commonly referred to as the "additive feature volume" V_{F_a} and the "subtractive feature volume" V_{F_s}, respectively. This is due to the fact that any additive form features will be subsets of V_{F_a} and any

[1] When necessary, the VFR theory as presented by Coles [11] will be referred to as the "original VFR."

subtractive form features will be subsets of V_{F_s}. The term "feature volume" is used to refer to the volume from which form features will be recognized. "Feature volume" can refer to V_{F_a} or V_{F_s}, though it may refer to another volume entirely.



(a) V_P or V_{F_A} (b) V_S

(c) V_Δ or V_{F_S}

Figure 3 Basic volume types

All feature recognition techniques with a V_{F_a} and V_{F_s} (i.e., they generate additive or subtractive form features). Several techniques deal with additive *and* subtractive features, but this is done using different definitions for additive and subtractive features. However, this is inconsistent with the ideal of a single entity known as a "feature." Later methods, including the original VFR, improved upon this by using a single definition that defined additive or subtractive features depending on whether the technique was applied to V_{F_a} or V_{F_s}, respectively. The VFR theory uses these various types of volumes to simultaneously recognize additive and subtractive features.

2.1.2 Full extensions and connected extensions

The utility of extending geometry was introduced in an earlier section (see Figure 1). In order to rebuild the geometry that may have been obscured by intersecting form features, the faces of the feature volume are extended. In effect, this is accomplished by calculating the intersections of the surfaces of each of the faces of the feature volume with the feature volume itself. This is defined as the **full extension** of a face *f* and can be symbolically defined as $ext(f) = S_f \cap V_F$, where S_f is the unbounded surface of a face *f* of the feature volume. However, the entire full extension of a face is not necessary to define the more complex VFR entities discussed in later sections. If the feature volume is concave, only that part of the full extension within the local convex portion of the volume is necessary. This portion of the full extension is the *connected* extension *c* of a face and is defined as follows:

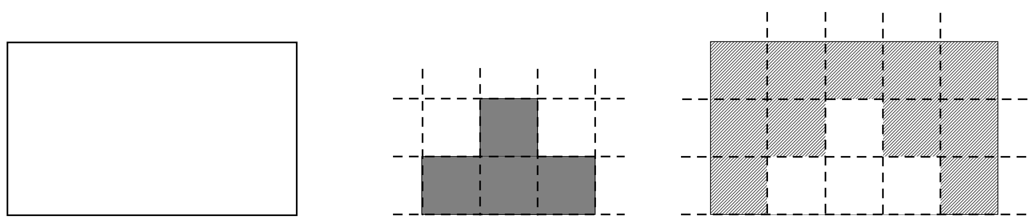
Definition 1 Let V_F be a feature volume of a part *P*. Let *f* be a face of V_F and S_f be the unbounded surface *f*. Finally, let *I* be the intersection of S_f and V_F, and let $I_{c,2m}$ be the set of all connected, 2-manifold subsets of I. The **connected extension**, c, of f with respect to V_F is the maximal element of $I_{c,2m}$ that is also a superset of f:

$$c \overset{\mathbf{max}}{\in} \{X \mid X \subset (S_f \cap V_F), 2-manifold_P(i(X)), Connected_P(X), X \supseteq f\}.$$

Notice that the connected extension of a face with no concave edges is identical to the face itself (i.e., $S_f \cap V_F$ is equal to f). This suggests a significance to the concave edges of the part. In fact, the concave edges truly separate one portion of the volume from another. If we extend those faces that have concave edges, we in effect define the boundaries between these distinct portions of the volume, and thus define the boundaries of the features of the volume. If the connected extensions are calculated relative to just additive feature volume V_{F_a}, only additive features are defined. If defined relative to just the subtractive feature volume V_{F_s}, only subtractive features are defined. This suggests that, in order to define both subtractive and additive features, when necessary, another sort of volume entirely is required.

2.1.3 Extended feature volume

Figure 4 illustrates a significant property of face extensions defined relative to the additive and subtractive feature volume. The unbounded surfaces of a significant portion of the faces that generate important connected extensions relative to each type of feature volume are, in fact, identical for both of the additive and subtractive feature volumes. This is an artifact of the way in which the feature volumes V_{F_a} and V_{F_s} are defined.[1]



(a) V_S (b) V_P (or V_{F_A}) (c) V_Δ (or V_{F_S}) with the same extensions

Figure 4 Surfaces are identical relative to V_{F_a} and V_{F_s}

This property suggests that a set of extensions might be calculated relative to a single volume that encompasses both the additive and the subtractive feature volumes and in doing so simultaneously define the boundaries of the additive and the subtractive features. A first suggestion for a volume that is include both V_{F_a} and V_{F_s} might be the stock volume V_S. However, implicit in this suggestion is the assumption that $V_S \supset V_P$. This is a common assumption in many feature recognition theories, and it holds true when the type of manufacturing being done is exclusively of subtractive nature (e.g., machining). However, it is not true generally. The invention of a new host of solid freeform fabrication methods that are primarily focused on the addition of part material (e.g., Stereolithography, Selective Laser Sintering) renders this implicit assumption incorrect. Another volume that is encompasses both V_{F_a} and V_{F_s} is an "extended" feature volume, defined as follows:

[1] Recall that V_{F_s} is equal to the delta volume $V_\Delta = V_S -^* V_P$.

Definition 2 Let V_P be a part volume and V_S the stock volume from which V_P will be manufactured. The **extended feature volume** V_{F_E} is equal to the boolean union of the stock volume and the part volume:

$$V_{F_E} = V_P \cup V_S.$$

A set of connected extensions calculated relative to V_{F_E} defines any additive and subtractive features necessary to describe the manufacture of the part volume from the stock volume. Figure 5 illustrates this using the entities shown earlier in Figure 4.

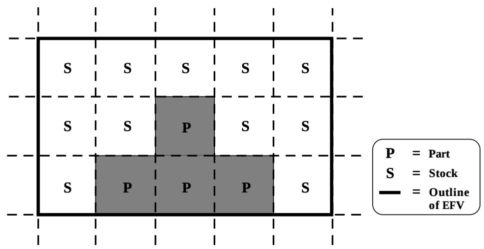
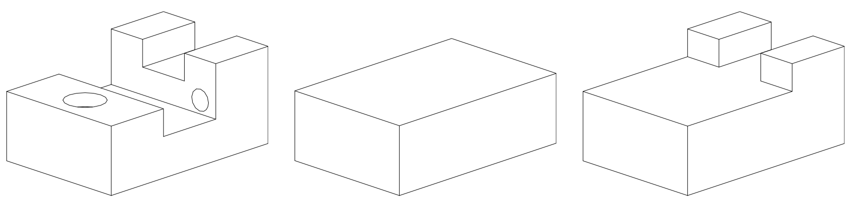


Figure 5 Defining an extended feature volume.

Figure 6 shows the example part, an example stock, and the extended feature volume defined using those volumes. Figure 7 displays the set of connected extensions of the extended feature volume shown in Figure 6.



(a) V_P (b) A sample V_S (c) V_{F_E}

Figure 6 Extended feature volume of the example part.

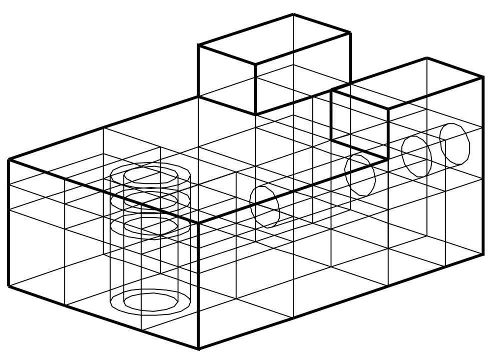


Figure 7 Connected extensions of example extended feature volume.

By defining the extended feature volume as it is above, the stock volume may be any volume at all, including a subset of the part volume or the empty set. If $V_S = \varnothing$, V_{F_E} will be equal to V_P and the features generated will be additive in nature. Likewise, if the stock volume is indeed a superset of the part volume, V_{F_E} will be equal to V_S and the features generated will be subtractive in nature. If, however, $V_S \not\subset V_P$ and $V_P \not\subset V_S$, the set of features generated will be a mix of additive and subtractive features. Most earlier feature recognition techniques are unable to accomplish this latter type of analysis. This is, however, a necessary analysis, because it reflects the true nature of manufacturing — for a given part volume, the manner in which it is manufactured is wholly dependent on where the process starts (i.e., the stock volume).

2.2 Decomposing to base volumes

From the connected extensions and feature volume are defined the "building blocks" of the features that the theory will later generate. These building blocks are known as "base volumes" and are formed by sub-dividing the extended feature volume using its set of connected extensions. As such, they are bordered on every side by a connected extension or a subset thereof. These portions of connected extensions that bound base volumes are known as "sub-faces" and are defined below.

2.2.1 Definition of a sub-face

The borders of base volumes are portions of connected extensions known as "sub-faces." The term "sub-face" is somewhat of a misnomer, as a sub-face may or may not be a subset of an actual face of V_P, V_S, or V_{F_E}. A sub-face is, in fact, a portion of a connected extension of a face that bounds a base volume. Due to the nature of connected extensions, the set of points that makes up a sub-face may in fact be entirely within the interior of V_{F_E}. Each connected extension (and thus each face of the part volume) defines a set of sub-faces as given below:

Definition 3 Let V_{F_E} be an extended feature volue with a connected extension c_i that is a member of the set of all connected extensions, $C = \{c_1, \ldots, c_n\}$. Let I_i be the set of curves calculated by intersecting c_i with every element of *C* except itself, resulting in a set of 1-manifold intersection graphs. A **sub-face** f_{sub} of V_{F_E} is the closure of *Y* (written *c*(*Y*)), where *Y* is a maximal element of the set of all connected subsets of $c_i - \bigcup_{k=1}^{m}(I_i)_k$, where $\bigcup_{k=1}^{m}(I_i)_k$ is the distributed union of all elements of I_i. Symbollically,

$$f_{\text{sub}} = c(Y), Y \overset{\textbf{max}}{\in} \{X \mid X \subset (c_i - \bigcup_{k=1}^{m}(I_i)_k), Connected_P(X) \mid I_i = \bigcup_{\substack{j=1 \\ j \neq i}}^{n}(c_i \cap c_j)\}.$$

The set of all sub-faces of V_{F_E} is written as **f** and is equal to the union of all of the sets of sub-faces calculated from each of the members of the set of connected extensions *C* of the extended feature volume.

2.2.2 Types of sub-faces

A "sub-face" may or may not be a subset of a face of V_{F_E}, V_P, or V_S, but it *will* be a subset of a connected extension of a face of V_P. This suggests that sub-faces can be classified by several different types. A particular sub-face may be classified by one or more of the following designations: "stock,", "part," "external," "internal," "real," or "virtual."

- The set of *stock* sub-faces (f_s) consists of those members of **f** that are subsets of some face *f* which is a part of the boundary of the stock volume V_S (written $b(V_S)$).

Symbolically,

$f_s = \{X \mid X \in \mathbf{f}, X \subseteq f \mid f \subset b(V_S)\}$.

- The set of *part* sub-faces (f_p) is made up of those members of **f** that are subsets of some face *f* of the part volume V_P. Symbolically, $f_p = \{X \mid X \in \mathbf{f}, X \subseteq f \mid f \subset b(V_P)\}$.

- The set of *external* sub-faces (f_e) consists of those members of **f** that are subsets of some face *f* of the extended feature volume V_{F_E}. Symbolically, $f_e = \{X \mid X \in \mathbf{f}, X \subseteq f \mid f \subset b(V_{F_E})\}$.

- The set of *internal* sub-faces (f_i) is made up of all members of **f** that are not members of f_e. Symbolically, $f_i = \{X \mid X \in \mathbf{f}, X \notin f_e\}$.

- The set of *real* sub-faces (f_R) are those members of **f** that are necessary to define the boundary of the part volume V_P. In fact, $f_R = f_p$.

- The set of *virtual* (f_V) contains the members of **f** that are also members of f_s but not members of f_p. Symbolically, $f_V = \{X \mid X \in f_s, X \notin f_p$.

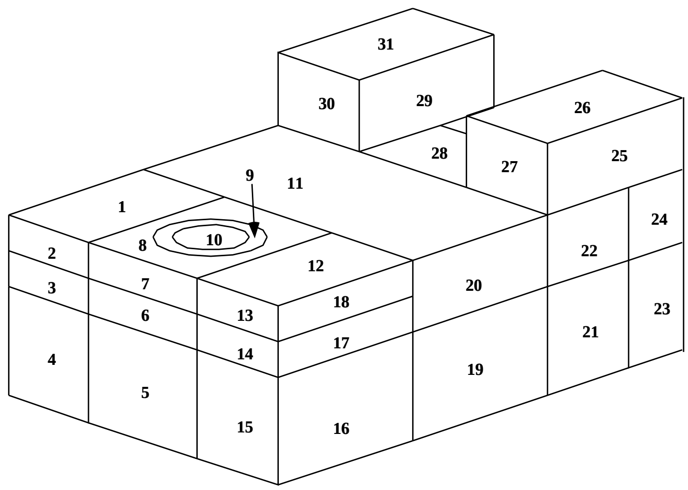


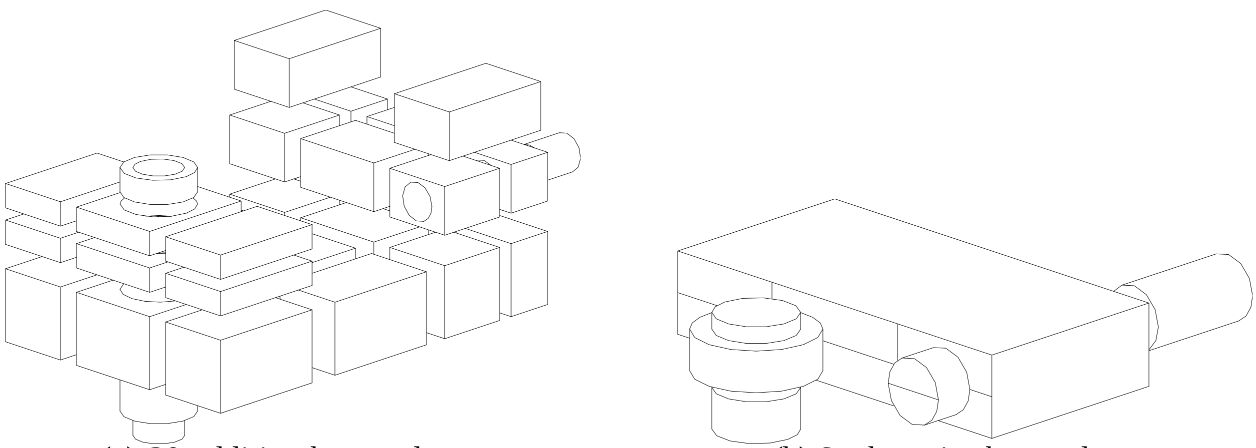
Figure 8 Some sub-faces of the sample extended feature volume

2.2.3 Definition of a base volume

Having defined the entities that make up the border of a base volume, we are now prepared to define the base volume itself. It is convenient to define the base volume in terms of the connected extensions themselves, though we know from Section 2.2.1 that the border of a base volume is made up entirely of members of **f**.

Definition 4 Let V_{F_E} be an extended feature volume with a set of connected extensions $C = \{c_1, \ldots, c_n\}$. A **base volume** V_b is the closure of Y (written $c(Y)$), where Y is a maximal element of the set of connected subsets of $V_{F_E} - \bigcup_{i=1}^{n} c_i$, where $\bigcup_{i=1}^{n} c_i$ is the union of all elements of C. Symbolically,

$$V_b \in \{c(Y) \mid Y \overset{\max}{\in} X \mid X \subset (V_{F_E} - \bigcup_{i=1}^{n} c_i), Connected_P(X)\}.$$



(a) 30 additive base volumes (b) 6 subtractive base volumes

Figure 9 Base volumes of the example extended feature volume

Figure 9 illustrates the base volumes of V_{F_E} for the sample part.

2.3 Recomposing into maximal simple features

Having defined base volumes, the building blocks from which features are constructed, we can now define an entity known as a *maximal simple feature*. The form features are recognized directly from maximal simple features.

2.3.1 Definition of a maximal simple feature

A maximal simple feature (MSF) is a maximal, convex subset of the feature volume that is bounded by a portion of a subset of the set of connected extensions. Formally:

Definition 5 Let V_{F_E} be an extended feature volume with a set of connected extensions $C = \{c_1, \ldots, c_n\}$. A **maximal simple feature** F_{MS} of V_{F_E} is a maximal element of the set of all V satisfying the following conditions:

1. V is a subset of $V_P \oplus V_S$,[1]

[1] $A \oplus B$ denotes the "exclusive or" operation, defined as $(A \cup B) - (A \cap B)$.

2. V is a simple feature, i.e., every edge e of V is convex relative to V,

3. the boundary of V is a subset of the distributed union $\bigcup_{i=1}^{n}(C_{sub})_i$ of the n members of $C_{sub} \subseteq C$.

Symbolically,

$$F_{MS} \overset{\mathbf{max}}{\in} \{V \mid V \subset (V_P \oplus V_S), Simple_P(V), \exists C_{sub} \subseteq C \mid b(V) \subseteq \bigcup_{i=1}^{n}(C_{sub})_i\}.$$

The following two propositions state that every total feature volume can be completely decomposed into a unique set of MSFs:

Proposition 1 (Uniqueness and Completeness) There is one and only one finite set of maximal simple features $\{F_{MS_1}, \ldots, F_{MS_n}\}$ of a semi-analytic total feature volume V_T.

Proposition 2 (Completeness) Given a total feature volume V_T with maximal simple feature set $\{F_{MS_1}, \ldots, F_{MS_n}\}$:

$$V_T = \bigcup_{i=1}^{n} F_{MS_i}.$$

Figure 10 shows the maximal simple features of the sample part from Figure 2.

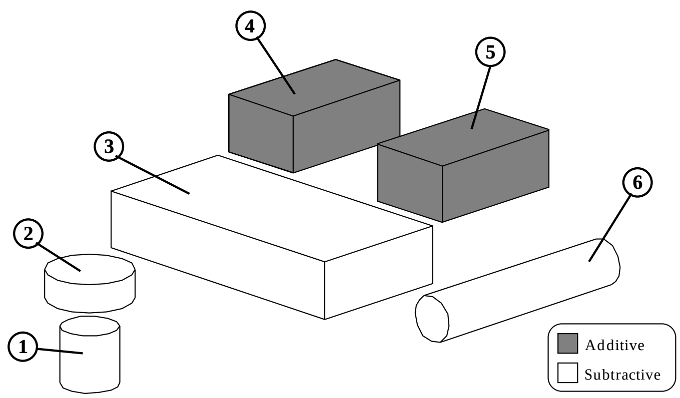


Figure 10 Maximal simple features of the example extended feature volume

While a maximal simple feature can be defined exclusively in terms of connected extensions, it is useful to think of maximal simple features as groups of base volumes.

Proposition 3 Given a total feature volume V_{F_E} with base volume set M_V, every maximal simple feature F_{MS} of V_{F_E} is the distributed union of some subset $(M_V)_{sub}$ of V_{F_E}:

$$\forall F_{MS}, \exists (M_V)_{sub} \subset M_V \mid F_{MS} = \bigcup (M_V)_{sub}.$$

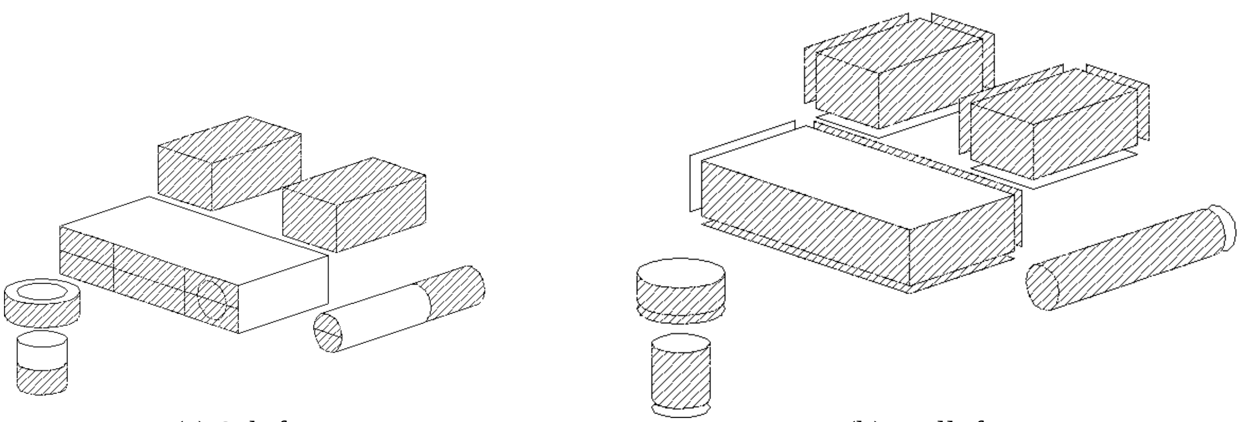
*Proof.*From the definition of a maximal simple feature (Definition 5), we know that F_{MS} must be bounded by subsets of connected extensions on all of its sides. Consider then that $F_{MS} \neq \bigcup (M_V)_{su}$. This requires that F_{MS} is a volume that is contains only a fraction of at least one base volume V_b. For this to be the case, and for F_{MS} to be bounded by a subset of a connected extension on all sides, a subset of a connected extension must exist in the interior of the base volume V_b which is only partially represented within F_{MS}. However, this is not consistent with the definition of a base volume (Definition 4). If a subset of a connected extension exists within a base volume, then $V_b - \bigcup_{i=1}^{n} c_i$ must necessarily divide V_b into at least *two* connected subsets. Yet a connected subset is precisely the definition of a base volume. Thus, $V_b - \bigcup_{i=1}^{n} c_i$ divides V_b into two distinct base volumes, causing F_{MS} to indeed be made up of only entire base volumes.❐

2.3.2 Classifying faces of a maximal simple feature

A quick note detailing the face classification of maximal simple features. Once a maximal simple feature is contructed from the distributed union of all the members of its set of base volumes M_V, each of the faces of the resulting volume can be classified according to the following rules: if a face *f* of maximal simple feature F_{MS} is a superset of one or more real sub-faces f_{sub}, then *f* is considered a *real* face. If *f* is a superset of no real sub-faces, then *f* is a *virtual* face.

This can be expressed another way: *real* faces are those necessary to define the boundary of the part volume; *virtual* faces are those faces that are not necessary to define the boundary of the part volume. Note that the definition of a virtual "sub-face" and a virtual "face" differs slightly. A virtual "face" might be a superset of only "internal" sub-faces; however, as long as no "real" sub-faces are subsets of the face, it is designated as "virtual."

This same classification scheme may also be used for other types of topology, namely edges. A *real* edge is one necessary to define the boundary of the part volume; a *virtual* edge is unnecessary for that same task.



(a) Sub-faces (b) Full faces

Figure 11 Faces of the maximal simple features

2.4 Defining features

Using maximal simple features, as defined in Section 2.3, we are at last able to define form features themselves. In doing so, we directly link (via connected extensions, base volumes, and maximal simple features) the form features of a part with its geometry. Note that the VFR method defines form features *solely* in terms of geometry, unlike many methods which begin the recognition process with an *a priori* set of features form which the process will search.

Form features are classified as *simple* or *compound*, and can be further classified as *additive* (consisting only of additive base volumes), *subtractive* (consisting only of subtractive base volumes), and *hybrid* (consisting of a mixture of additive and subtractive base volumes). These classifications are defined below.

2.4.1 Simple features

A simple feature is the most basic building block that is assigned semantic meaning. It is defined as follows:

Definition 6 Let F be a feature of the extended feature volume V_{F_E}. Furthermore, let F_{MS} be a maximal simple feature of V_{F_E} with a set of base volumes M_V. Feature F is a **simple feature** (written F_S) if the following conditions are met:

1. F is equal to the union of a set of base volumes **B**, where $\mathbf{B} \subseteq M_V$,

2. F is a connected volume,

3. all edges of F are convex with respect to F.

Symbolically,

$$F_S \in \{F \mid F = \bigcup_{i=1}^{n} \mathbf{B}_i \mid \mathbf{B} \subseteq M_V, Connected_P(F), Convex_P(F)\}.$$

As a corollary to Definition 6, a simple feature F is defined to be equal to a maximal simple feature F_{MS} or a convex subset thereof (i.e., $F \subseteq F_{MS}$).

2.4.2 Classifying simple features

To this point, we have defined an entity known as a "simple" form feature, but it is not yet clear how these entities map to the semantic information that makes form features such a powerful means of information transfer. This section presents a methodology for linking form features with semantic information. The means by which this information is linked to the features is via a Labeled Topological Adjacency Graph (LTAG). Properties of the faces of the simple feature volume are used to designate the nodes of the graph. The LTAG is then matched against a set of reference LTAGs that are linked to feature labels that imply semantic information (e.g., "hole," "slot").

2.4.2.1 Face classification

Recall that, if a face *f* of the simple feature is a superset of any real sub-face $f_{sub} \in f_R$, then *f* is a real face. Otherwise, *f* is a virtual face. This provides one level of classification for each node of the LTAG of the feature. Yet all real (or virtual) faces of a feature are not equivalent semantically. For example, there is an obvious difference between the real face at the bottom of a blind hole and the real face on the side of a blind hole. A set of structural primitives from the Intermediate Geometry Language (IGL) [14, 35] captures this distinction and provides the second level of classification for the LTAG nodes.

A face of a feature volume can be assigned one of four structural primitives. They are **TOP**, **BOTTOM**, **SIDE**, and **END**. Detailed definitions of all the IGL primitives are available in [14]. For the purposes of the VFR method, there are two important facts in regards to the structural primitives:

I. the primitives are defined using geometric and topological entities (i.e., no *a priori* information is required to assign the primitives); and

II. the primitives are defined topologically to be identical for additive and subtractive features.

As these two properties are shared by the VFR method (vis-à-vis its definition of features), the structural primitives integrate well into the VFR theory.

2.4.2.2 Constructing the labeled topopology adjacency graph

Using the attributes defined for faces and edges, a Labeled Topology Adjacency Graph (LTAG) is constructed for a given feature. The LTAG shows the relationships between the faces and edges of the feature volume along with information regarding the face type (e.g., "top," "bottom"). Figure 12 illustrates the LTAGs for a BLIND HOLE, a THROUGH HOLE and a SLOT feature.

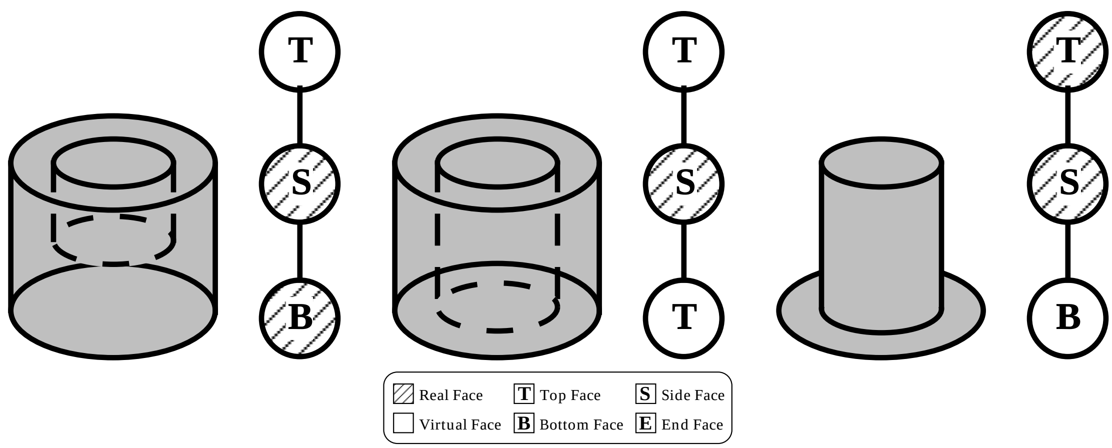


Figure 12 LTAGs of BLIND HOLE, THROUGH HOLE, and BOSS features

The feature LTAG is then compared for Labeled Face Edge Topology (LFET) equivalence (written $A \overset{LFET}{=} B$) to a reference feature defined in a reference hierarchy. Lest the reader draw incorrect parallels between matching feature LTAGs against a reference hierarchy and the graph matching methods discussed in the introduction, it is important to point out that the *definition* of the form feature has already occured at this point in the VFR method. The classification against a reference set of features is necessary only to assign a semantic label to the feature. No matter what semantic label we assign to this feature or that, the actual definition of the feature remains unchanged . This is in stark contrast to graph matching methods that in fact *define* a "feature" based upon the reference hierarchy.

Divorcing the feature definition (and thus the feature recognition) from the feature classification allows for a powerful type of analysis. Namely, the same set of recognized features can be matched against several different feature classification hierarchies depending upon the context. For instance, the features can be recognized once, classified by a design engineer for design analysis, and then later classified by a manufacturing engineering for manufacturing analysis.

2.4.3 Relationships between simple features

By recognizing and classifying simple features, we possess the ability to make a reasonable geometric analysis of a part. However, by adding a second level of abstraction — namely, by characterizing the relationships *between* simple features — we can gain further insight into the feature description of the part. The benefit of this approach is made clear by example. Consider a part volume with two features, a HOLE feature and a SLOT feature. The manufacturability of the part is impacted significantly if the HOLE intersects the bottom of the SLOT instead of the side of the SLOT.

Again, we are able to rely on primitives defined by the IGL. The IGL defines a set of relationships between features in a topologically independent manner. The IGL concepts are summarized below and, alternative definitions are given in sections that follow.

2.4.3.1 Interacting relationships

Interacting relationships are defined as those feature relationships where the features physically interact with one another. Generally, two features may physically intersect with a two-dimensional point set or a three-dimensional point set.[1] A relationship type is defined for each of these cases.

2.4.3.1.1 ABUT relationship

Let F_1 and F_2 be two distinct simple features with sets of bordering sub-faces M_{F_1} and M_{F_2}. F_1 **ABUT**s F_2 if and only if there exists a sub-face f_{sub} that is a member of the bordering face set of both features. Symbolically,

$$F_1 \mathbf{ABUTS} F_2 \Leftrightarrow \exists f_{sub} \mid f_{sub} \in M_{F_1}, f_{sub} \in M_{F_2}.$$

2.4.3.1.2 INTERSECT relationship

Let F_1 and F_2 be two distinct simple features made up of sets of base volumes M_{V_1} and M_{V_2}, respectively. M_{V_1} **INTERSECT**s M_{V_2} if and only if there exists at least one base volume V_b that is a member of the base volume set of both features. Symbolically,

$$F_1 \mathbf{INTERSECTS} F_2 \Leftrightarrow \exists V_b \mid V_b \in M_{V_1}, V_b \in M_{V_2}.$$

Worthy of note is the fact that two simple features cannot **ABUT** or **INTERSECT** one another unless the maximal simple features from which they were formed also **INTERSECT** one another (i.e., $M_{V_1} \cap M_{V_2} \neq \emptyset$).

2.4.3.2 Interfeature relationships

An interfeature relationship is one in which a relationship exists between two features but the features do not necessarily interact physically with one another (as they do in interacting relationships). Interfeature relationships come in two types -- those that relate feature faces, and those that relate feature axes.

2.4.3.2.1 PLANAR relationship

Let F_1 and F_2 be two distinct simple features. Furthermore, let f_1 be a face of F_1 with connected extension c_1, and f_2 be a face of F_2 with connected extension c_2. F_1 is **PLANAR** with F_2, with respect to f_1 and f_2 if f_1 and f_2 are virtual faces and $c_1 = c_2$.

---

[1] Features could intersect with a one-dimensional point set, but this arguably is not a “significant'“ physical interaction.

2.4.3.2.2 COPLANAR relationship

Let F_1 and F_2 be two distinct simple features. Furthermore, let f_1 be a face of F_1 with full extension $ext(f_1)$ and connected extension c_1, and f_2 be a face of F_2 with full extension $ext(f_2)$ and connected extension c_2. F_1 is **COPLANAR** with F_2, with respect to f_1 and f_2 if f_1 and f_2 are virtual faces, $ext(f_1) = ext(f_2)$ and $c_1 \neq c_2$.

2.4.3.2.3 OFFSET relationship

Let F_1 and F_2 be two distinct simple features. Furthermore, let f_1 be a face of F_1 with connected extension c_1, and f_2 be a face of F_2 with connected extension c_2. F_1 is **OFFSET** from F_2, with respect f_1 and f_2 if f_1 and f_2 are virtual faces, $c_1 \neq c_2$, but the face normal of f_1 is equal to the face normal of f_2.

The next set of relationships makes use of "feature axes." A feature axis, for most features, is along the access direction of the tool for machining. While the IGL does not present a general definition for a feature axis, it is relatively easy to suggest one for different types of features. For prismatic features, the feature axis may be defined as follows: if a feature has two topologically parallel virtual faces, the axis is along the line passing through the centroids of these two faces. If the feature does not have two topologically parallel virtual faces, the axis is along the line passing through the centroids of a virtual face and a real face that are topologically parallel. For cylindrical features, the feature axis is defined to be the axis of symmetry. Some features such as the re-entrant and open pocket contain multiple axes. To resolve ambiguity over which axis contributes to a particular relatonship, the axis is also made part of the relationship specification.

2.4.3.2.4 PARALLEL relationship

Let F_1 and F_2 be two distinct simple features with axes $\vec{a}_1$ and $\vec{a}_2$, respectively. F_1 is **PARALLEL** to F_2, with respect to $\vec{a}_1$ and $\vec{a}_2$, if the angle between $\vec{a}_1$ and $\vec{a}_2$ is 0 or 180 degrees.

2.4.3.2.5 COLINEAR relationship

Let F_1 and F_2 be two distinct simple features with axes $\vec{a}_1$ and $\vec{a}_2$, respectively. F_1 is **COLINEAR** to F_2, with respect to $\vec{a}_1$ and $\vec{a}_2$, if the angle between $\vec{a}_1$ and $\vec{a}_2$ is 0 or 180 degrees, and the direction of $\vec{a}_1$ is equal to the direction of $\vec{a}_2$.

2.4.3.2.6 ORTHOGONAL relationship

Let F_1 and F_2 be two distinct simple features with axes $\vec{a}_1$ and $\vec{a}_2$, respectively. F_1 is **ORTHOGONAL** to F_2, with respect to $\vec{a}_1$ and $\vec{a}_2$, if the angle between $\vec{a}_1$ and $\vec{a}_2$ is 90 or 270 degrees.

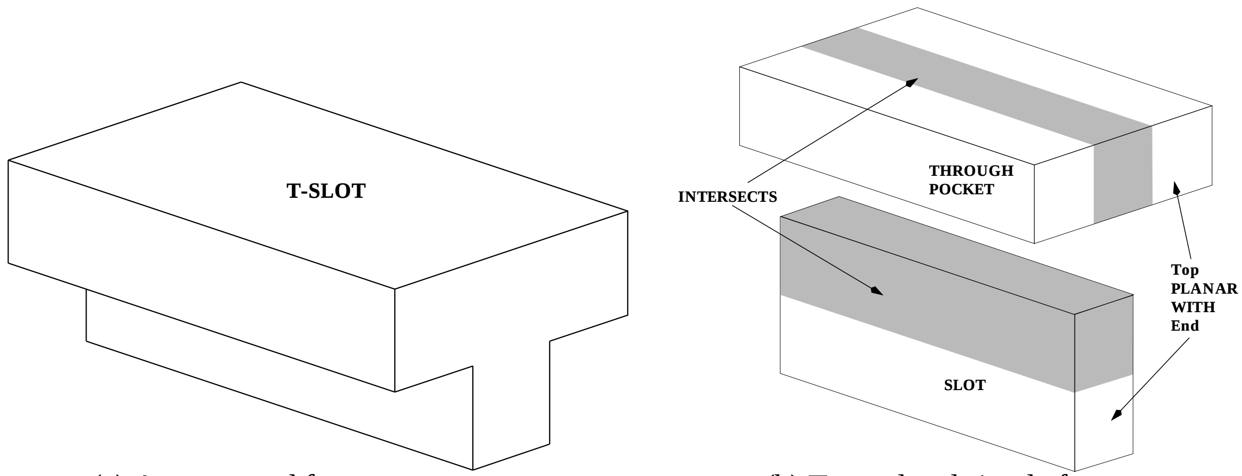
2.4.3.2.7 ANGULAR relationship

Let F_1 and F_2 be two distinct simple features with axes $\vec{a}_1$ and $\vec{a}_2$, respectively. F_1 is **ANGULAR** to F_2, with respect to $\vec{a}_1$ and $\vec{a}_2$, if the angle between $\vec{a}_1$ and $\vec{a}_2$ is not 0, 90, 180, or 270 degrees.

2.4.4 Defining compound and hybrid features

Simply put, a compound feature is any feature that is not a simple feature. By implication from Section 2.4.1, it is clear that a compound feature is *not* a subset of a single maximal simple feature and is not necessarily a convex volume. The simple features defined in Section 2.4.1 and the feature relationships defined in Section 2.4.3 provide the basis by which we can define compound feature.

A **compound feature** F_C is defined as two or more simple features related via an interacting relationship *and* an interfeature relationship (as defined in Section 2.4.3). For example, a simple feature HOLE1 that is **COLINEAR** with and **INTERSECTS** a second simple feature HOLE2 may be defined as a compound feature type COUNTERBORE. Just as simple features are, in effect, groups of base volumes, compound features are groups of relationships between simple features. Figure 13 illustrates an example of a compound feature.

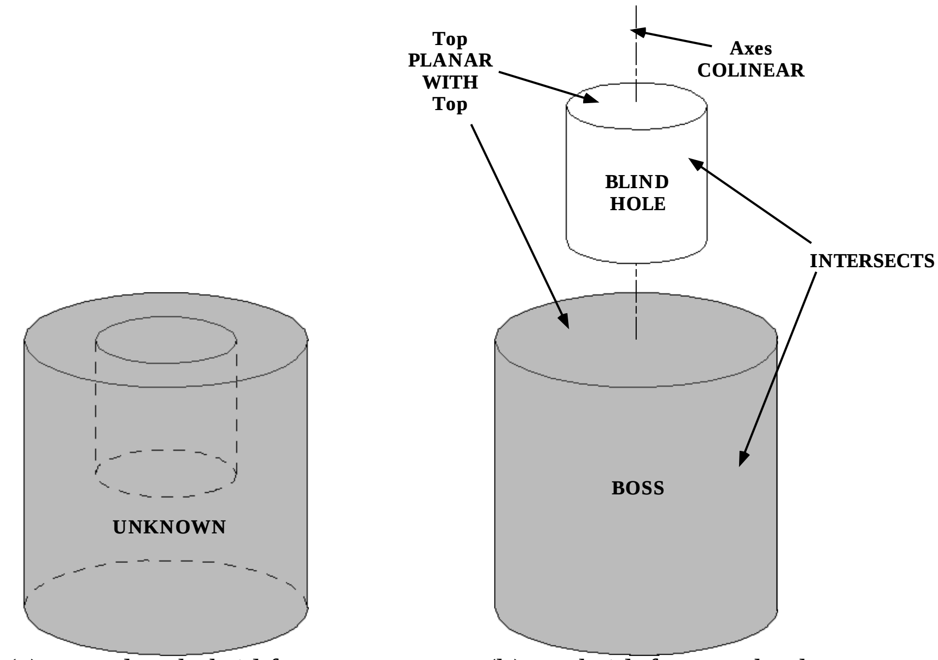


(a) A compound feature (b) Two related simple features

Figure 13 An example of a compound feature

2.4.4.1 Hybrid features

A **hybrid feature** is a specialized sub-type of a compound feature. It is a compound feature whose set of relationship types relate simple features of mixed types (i.e., additive and subtractive). A hybrid feature is manufactured by first creating a feature of one type, and then creating the feature of the other type. In contrast, a general compound feature such as a COUNTERBORE might be made with one manufacturing operation. Figure 14 illustrates an example of a hybrid feature. The nameless hybrid feature (shown in Figure 14(a)) is equal to an additive BOSS feature that **INTERSECTS** a subtractive BLIND HOLE feature, and whose top face is **PLANAR** with the top face of the hole (shown in Figure 14(b)).



(a) Nameless hybrid feature (b) Hybrid feature broken out

Figure 14 An example of a hybrid feature

2.5 Defining feature sets

To this point, we have defined maximal simple features, simple features, and compound features. This section discusses how multiple sets of simple and compound features are generated from a group of maximal simple features. A **feature set** is a set of n features $\{F_1, \ldots, F_n\}$ defined such that the union of the set of features and the stock volume equals the part volume. Symbolically, $V_P = V_S \bigcup_{i=1}^{n} F_i$. Feature sets are calculated from the maximal simple features using the method outlined below.

2.5.1 Generating feature sets from feature sequences

A feature sequence is defined as a distinct ordering of the set of maximal simple features. For example, from the six maximal simple features of our example part (shown in Figure 10), one possible feature sequence is {1, 3, 2, 5, 4, 6}. For each feature sequence of maximal simple features, a set of simple features can be generated by the following method:

> For a given ordered set of n maximal simple features $\{F_{MS_1}, \ldots, F_{MS_n}\}$, begin with the first maximal simple feature of the sequence F_{MS_1}. Create a simple feature from the union of all the base volume members of M_{V_1} of F_{MS_1}. For every other maximal simple feature $F_{MS_j}, j > i$ in the sequence, create one or more simple features equal to the connected subsets of the union of all the base volume members of M_{V_j}, excepting those base volumes that are members of M_{V_k} of any $F_{MS_k}, k < j$ considered earlier in the sequence. Continue the process until all n maximal simple features in the sequence are considered. This generates a set of at least n simple features.

For each simple feature in the set, a LTAG is calculated and matched against the set of reference feature LTAGs. This results is a set of "semantically labeled" simple features. Figure 15 shows one valid feature set for the example part.

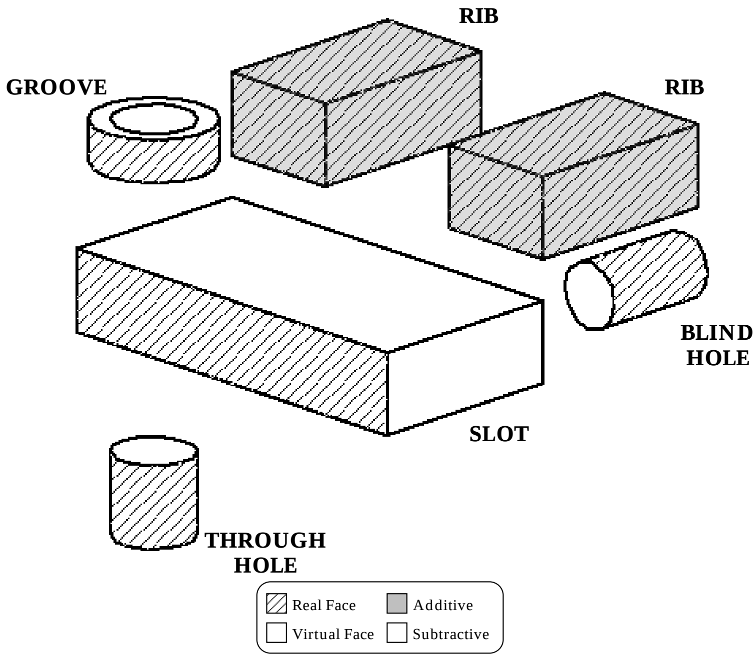


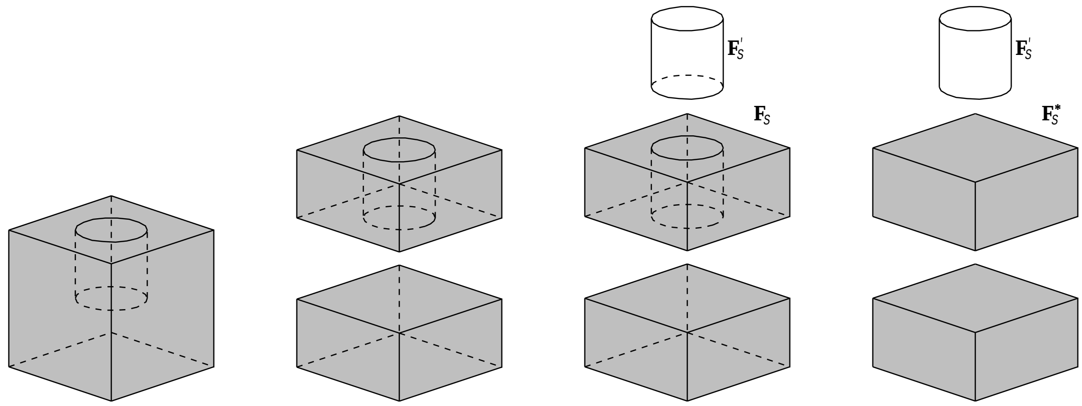
Figure 15 One feature set of the example part

The next step is to find potential compound features. This is accomplished by examining the relationships each simple feature in the set has with the other simple features. These relationships (e.g., HOLE1 is **COLINEAR** with HOLE2 and the bottom of HOLE1 **ABUTS** the top of HOLE2) are also matched against the set of reference features to see if a corresponding compound feature (e.g., COUNTERBORE) is present. If such a feature exists, a separate feature set is created with the compound feature.

To find hybrid features (a subset of compound features), each of the simple features in the set is examined using the following methodology:

Let F_S be a simple feature with a set of base volumes M_V and a set of bordering extensions M_E. And let M'_V be a second set of base volumes with a set of bordering extensions M'_E, where the base volumes in M'_V are the opposite type of the base volumes in M_V. If $M'_E \subset M_E$, then a potential hybrid feature exists that is equal to the union of M_V and M'_V, and that has an **INTERSECTS** relationship with the union of the members of M'_V.

Each of the resultant simple features are then classified against the reference hierarchy and the resultant SIMPLE FEATURE **INTERSECTS** SIMPLE FEATURE is matched against the list of reference compound features. Figure 16 illustrates the process of recognizing a hybrid feature.



(a) Original part (b) 2 additive features (c) F_S and potential F_S' (d) 1 hybrid and 1 additive feature

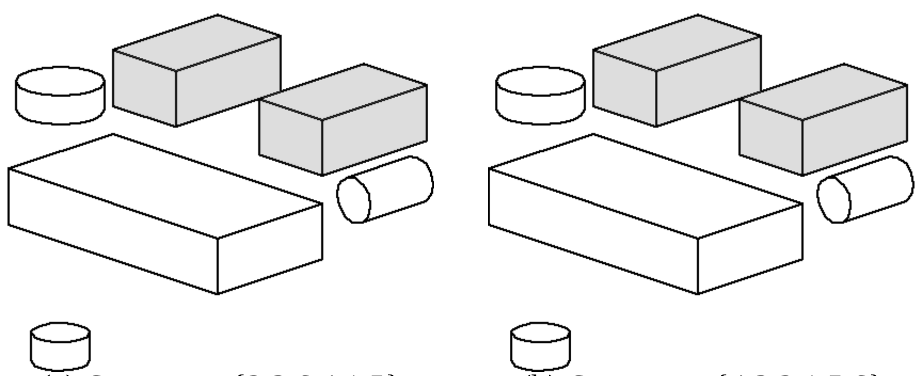
Figure 16 Recognizing a hybrid feature

2.5.2 Selecting feature sequences

The method described in Section 2.5.1 can be run on any valid feature sequence. For a set of *n* maximal simple features, *n*! distinct sequences exist. Just from the six maximal simple features of the example part, 720 different sequences can be used to create feature sets. The number quickly grows out of control for larger sets of maximal simple features.

Computational complexity is an admitted problem with feature recognition methods based on surface extension. However, it is important to recognize that, although large numbers of feature sets are generated for a given part, the method is exhaustive. That is to say that it *will* generate *all* of the valid feature sets that describe the given part volume. Other methods, while they may not suffer from the computational complexity, limit themselves by their methodology and may in fact not recognize valid feature descriptions because of it.

The problem of computational complexity is not as dire as this worst case scenario. Consider the two feature sets shown in Figure 17. The two sets are created using different feature sequences, and yet result in identical feature sets. This is because distinct feature sets are defined *only by the relative order of maximal simple features that have intersecting relationships*. The relative order of those maximal simple features that do *not* intersect other maximal simple features is irrelevant.



(a) Sequence: {2,3,6,4,1,5} (b) Sequence: {4,3,2,1,5,6}

Figure 17 Equivalent feature sets

It can be shown that, rather than there being *n*! distinct feature sets created for a set of *n* maximal simple features, there are instead $m_1! \times m_2! \times \cdots \times m_{z-1}! \times m_z!$ distinct feature sets, where $m_1, \ldots m_z$ are equal to the numbers of the distinct maximal groups of maximal simple features that interact at any point. For example, consider the maximal simple features of the example part, as shown in Figure 10. There are two maximal groups of maximal simple features that interact: MSF 1 intersects MSF 2, and MSF 3 intersects MSF 6. This means there are $2! \times 2! = 4$ distinct feature sets. This is a vast improvement from 720. Figure 18 shows the four distinct feature sets that are calculated from the maximal simple features of the example part.

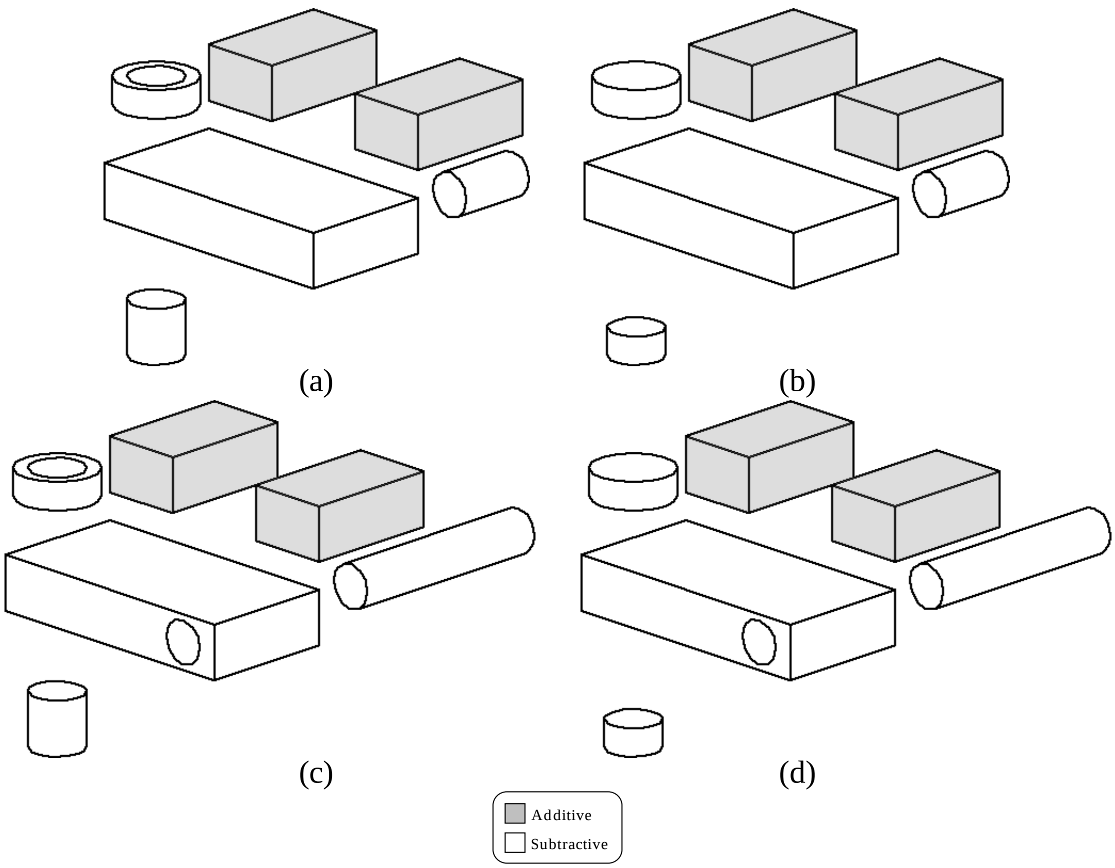


Figure 18 Distinct feature sets of the example part

It is important to note that the VFR theory is *not* concerned with process planning. If the method were generating all possible *processes* sequences that could be used to create the *n* maximal simple features, then in fact there *would* be *n*! distinct different sequences. However, the VFR theory is focused solely on recognizing distinct feature sets. Once those distinct feature sets are recognized, the sets can be transmitted to more sophisticated analysis methods that reason about the best process order necessary to create a given set of features.

However, despite this fact, the VFR method does provide valuable information that is of use in a process planning context. For example, consider the meaning of the real and virtual faces of a feature. If a feature has no virtual faces, then it is by definition inaccessible to any potential manufacturing method. This invalidates feature sets that contain such features. Figure 19 (a reproduction of Figure 18(d)) shows one such set. The designated feature has no virtual faces, meaning it is surrounded on all sides by faces that intersect the part boundary. This means it is inaccessible to manufacturing tools. A process planning application could use this sort of reasoning to discard certain feature sets generated by the VFR method.

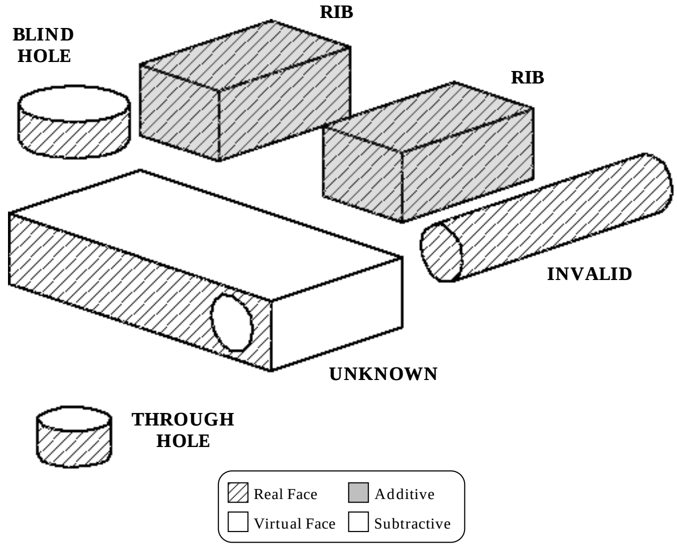


Figure 19 An invalid feature set.

Alternately, the set of feature sequences that generate distinct feature sets could be reduced further. If a maximal simple feature F_{MS} has no virtual faces, no valid feature set may be generated from a feature sequence in which F_{MS} appears *before* at least one maximal simple feature with which it has an **INTERSECT** relationship. If F_{MS} does not have an **INTERSECT** relationship with at least one other maximal simple feature, then the design of the part is, by definition, non-manufacturable (i.e., at least one of its features cannot be manufactured), and so the sequence can be discarded.

3. Making Engineering Judgements Using the Theory

This section presents three such case studies. The first considers a pivot sub-assembly and focuses on how the methodology can generate different feature interpretations from different stock volumes for a given part volume. The second case study focuses on how the methodology can return features that are useful from a design analysis standpoint. The third case study generates alternate feature sets for a base plate and analyzes those feature sets to determine which set will be the least costly to manufacture.

3.1 Case Study #1: Manufacturability Analysis Based on Different Stock Volumes

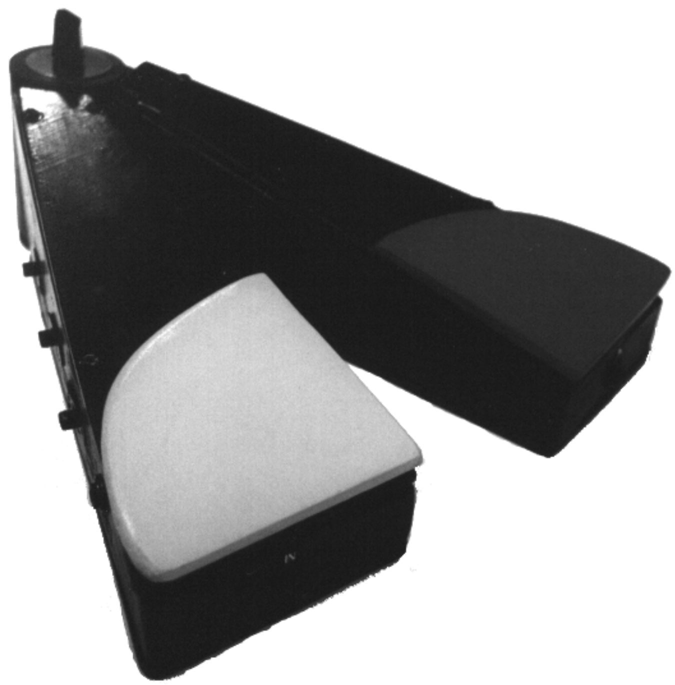
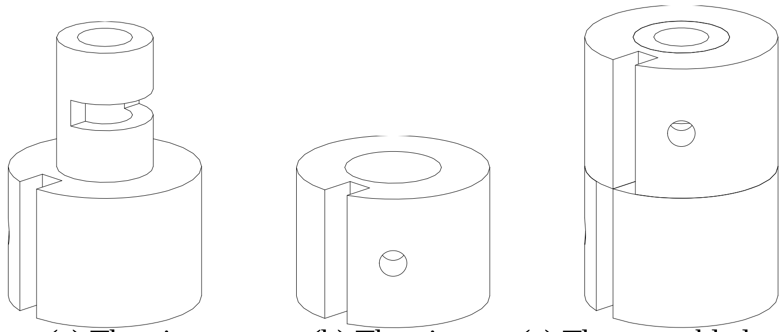


Figure 20 The Wingspeak

As explained in previous sections, a powerful ability of the theory is generating feature sets or descriptions relative to an appropriate stock volume. In other words, if it is given a stock volume larger than the part volume, the feature sets generated will be subtractive in nature. Likewise, a stock volume smaller than the part volume will drive the generation of additive feature sets. In all cases, the focus is on returning the features necessary to manufacture the part volume from the stock volume.

This section demonstrates that capability on a part of an device known as the WingSpeak. The WingSpeak (shown in Figure 20) was developed at the University of Texas at Austin and is a prototype of an assistive communication device for individuals with severe mental and/or physical disabilities. The specific part used in this case study is a portion of the pivot sub-assembly that enables electrical wiring to pass from one side to the other while still allowing the two sides to pivot relative to one another. Figure 20 shows the two sides of the WingSpeak slightly pivoted away from one another.

The pivot sub-assembly (shown closeup in Figure 21(c)) is simple, with wires passing from a side housing through a hole in one of the parts in the sub-assembly, through a cental axis, and then through the other part into the other side housing. The parts involved are simple in nature, having been designed to be manufactured in a relatively low-tech “wood shop” environment.



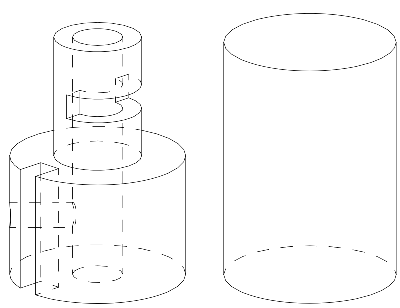
(a) The pivot lower half (b) The pivot upper half (c) The assembled pivot

Figure 21 The assembled Wingspeak pivot and lower joint

The analysis focuses on the specific part shown in Figure 21(a). The following analysis demonstrates how the theory presented in the previous sections can be used to suggest different manufacturing methods for the part. The theory can generate features for the low-tech way in which the part was manufactured in the "wood shop," as well as generate features for a more high-tech way in which the part would be manufactured in more high volume production.

3.1.1 Manufacturing from a larger stock

The first situation considered is if the features of the part V_P are generated relative to a larger stock volume V_S. Figure 22 shows these two volumes.



(a) V_P (b) V_S

Figure 22 Pivot and larger stock

Given that V_S is a superset of V_P, we know that the maximal simple features generated will be subtractive in nature. In fact, five subtractive maximal simple features are generated from the given V_S and V_P. Given that three sets of two features interact, $2! \times 2! \times 2! = 8$ distinct feature sequences exist with which to generate feature sets. One such feature set is chosen arbitrarily and shown in Figure 23.

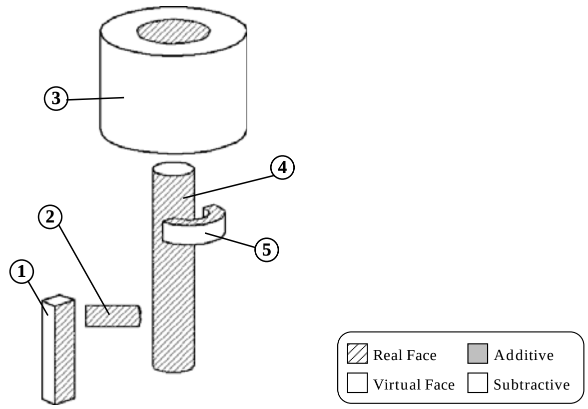


Figure 23 One possible set of features generated from larger stock

The numbers in Figure 23 represent the sequence number in which that particular feature will be manufactured. By following the sequence shown, V_S can be manufactured into V_P using five machining operations. Figure 24 shows the sequence of operations that generate the part volume (Figure 24(f)) from the stock volume (Figure 24(a)).

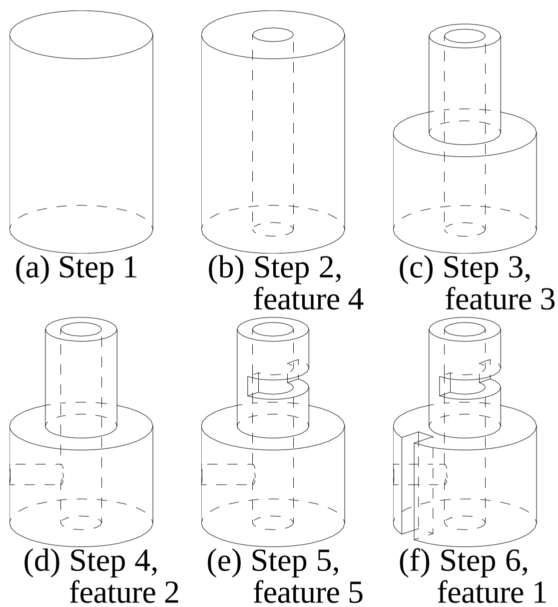
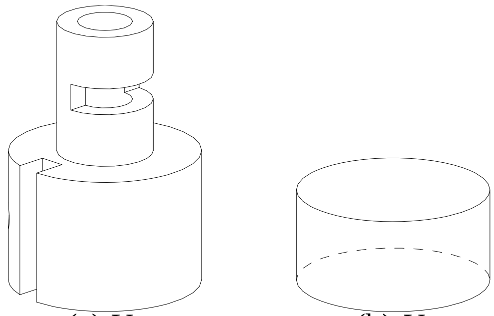


Figure 24 Creating the pivot using the features shown in Figure 23

3.1.2 Manufacturing from a smaller stock

Alternately, the part could be manufactured from a stock volume that is *not* a superset of the part volume. A potential stock volume is suggested naturally by the geometry of the part. A cylindrical stock volume that only encompasses the lower half of the part volume is a potential candidate from which to manufacture the part. Figure 25 shows the part and stock volumes used in this example.



(a) V_P (b) V_S

Figure 25 Pivot and smaller stock

By choosing a stock volume that is not a superset of the part volume, we necessitate a set of maximal simple features that is not homogeneously subtractive. In fact, of the 7 maximal simple features generated, 3 are additive and 4 are subtractive. Figure 26 shows one of the 32 distinct feature set that can be generated from the maximal simple features.

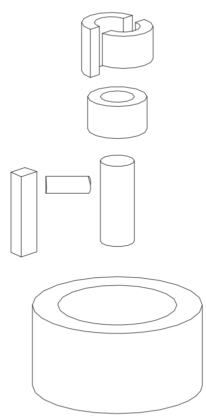
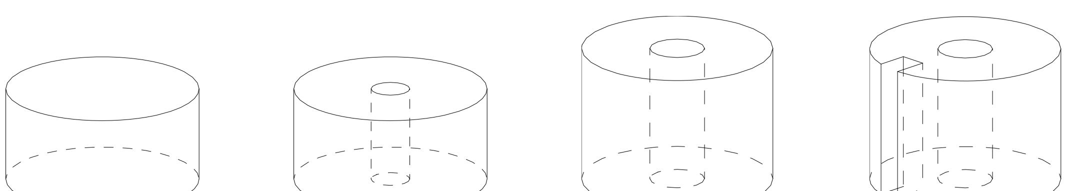
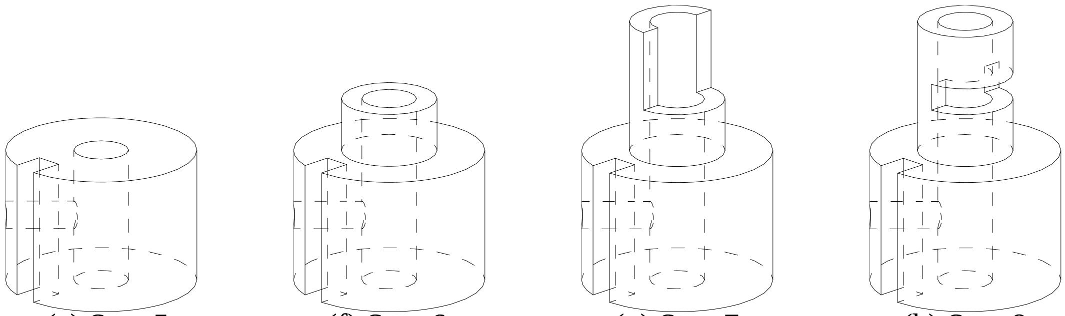


Figure 26 A set of features describing the pivot part

As before, the numbers in Figure 26 denote the sequence in which the features shown could be manufactured to create V_P from V_S. Using four subtractive manufacturing processes and three additive manufacturing processes, the part volume can be created from the stock volume. Figure 27 illustrates this manufacturing sequence from the stock volume (Figure 27(a)) to the part volume (Figure 27(h)).



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4



(e) Step 5 (f) Step 6 (g) Step 7 (h) Step 8

Figure 27 Creating the pivot using the features shown in Figure 26

While the sequence shown in Figure 27 faithfully creates the part volume using the features shown in Figure 26, a portion of the sequence is decidedly non-intuitive. Namely, steps 6 through 8 wherein material is additive to the intermediate work volume do not conform to what intuition tells us is the “proper” way to manufacture the part. A more intuitive set of features includes a *hybrid* feature (introduced in Section 2.4.4.1). While the base level VFR method does not explicitly incorporate hybrid features, the method *does* support such a representation, providing that an alternate algorithm is used to construct the maximal simple features. Refer to [1] for a detailed discussion of this alternate method. For now, Figure 28 shows a potential feature set using a hybrid feature that results in a more intuitive manufacturing process (as shown in Figure 29).

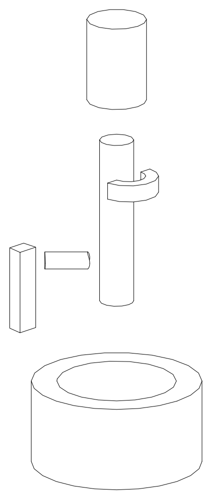
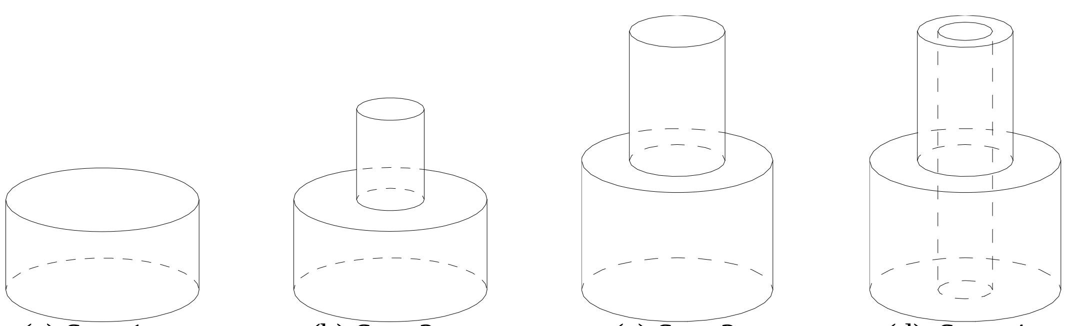


Figure 28 A feature set involving a hybrid feature



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

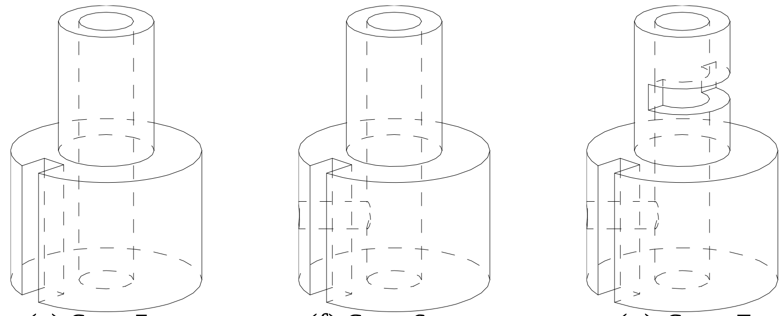

(e) Step 5 (f) Step 6 (g) Step 7

Figure 29 Creating the pivot using the features shown in Figure 28

3.1.3 Discussion of results

Hybrid features aside, what can be learned from the differences between the manufacturing method suggested by the feature set in Section 3.1.1 and the method suggested by the feature set in 3.1.2. Well, quite obviously, the two allow for a manufacturing analysis to be done based upon whatever stock volume is available. If the only stock available is one of the profile as in Figure 25(b), then the sequence presented in Section 3.1.2 is more useful.

However, the method also allows an engineer or engineering team to examine “what-if” situations with different potential stock volumes. At least, the analysis may suggest a method for manufacturing the part that the team had not considered. At best, the analysis might help to make a difficult choice between two seemingly similar manufacturing methods (especially if the analysis is taken once step further to explicitly consider cost, as is done with the case study in Section 3.3).

Of course, the means to manufacture a given feature set may not always be available. It is important for a feature recognition method to support manufacturability analysis that does not assume an “ideal” manufacturing environment. The VFR method, due to its ability to consider different sized stock volumes for a given part volume, enables this sort of “non-ideal” manufacturing analysis. Future work in VFR might include using a direct engineering database of available manufacturing processes to choose only feature sets that can be manufactured with specified processes. In addition to filtering unwanted information, this could considerably reduce the unfathomable combinations of feature sets.

3.2 Case Study #2: Analysis of Design Features

The case studies in Section 3.1 and Section 3.3 focus on performing an analysis of the manufacturability of a part using form features recognized from its boundary representation. While analyses of manufacturability are undoubtedly useful and necessary, by nature form features are flexible enough to represent many different types of semantic information[1]. One type of semantic information that features are well-suited to represent is information related to the *design* of a part.

This case study will focus on the connecting rod part represented in Figure 30. A connecting rod has several distinct features which can be directly linked to a functional design purpose. The challenge of this case study will be to demonstrate that the VFR method can isolate these design features from the boundary represetation of the part.

[1] See Section “A Study In Contrasts”

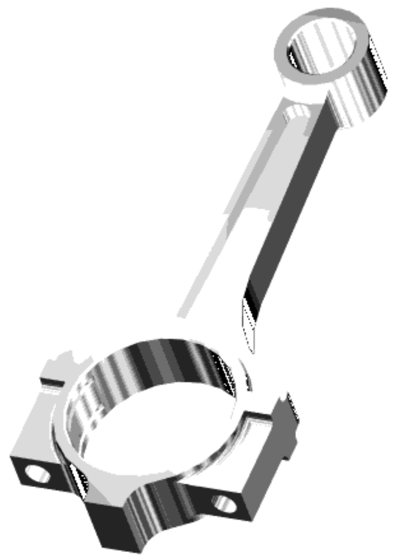


Figure 30 A connecting rod

3.2.1 Locating an appropriate feature set

For the purposes of the VFR method, our stock volume V_S is the empty set. We are not calculating feature relative to any stock volume, but are instead focusing on the features present on the part itself[1]. Following the VFR methodology for a empty stock volume results in a set of 11 maximal simple features, all of which are additive. From this set of 11 maximal simple features, many feature sequences result in distinct feature sets. One such set, of 15 additive features (some of the maximal simple features split into two distinct features), is illustrated below in Figure 31.

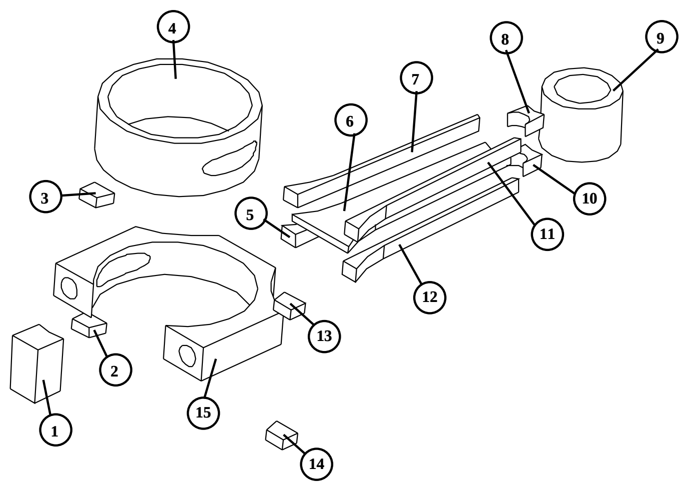


Figure 31 A feature set of the connecting rod

3.2.2 Analyzing the feature set

Upon closer analysis, it is clear that the features represented in Figure 31 have *some* significance, though exactly what we cannot ascertain solely from the figure. The volumes isolated as "features"

[1] Note that this yields a set of exclusively additive features of the part.

are volumetrically distinct sections of the boundary of the connecting rod. This suggests that they may serve some purpose within the overall function of the part. In fact, more detailed observations from a design perspective reveals just that.

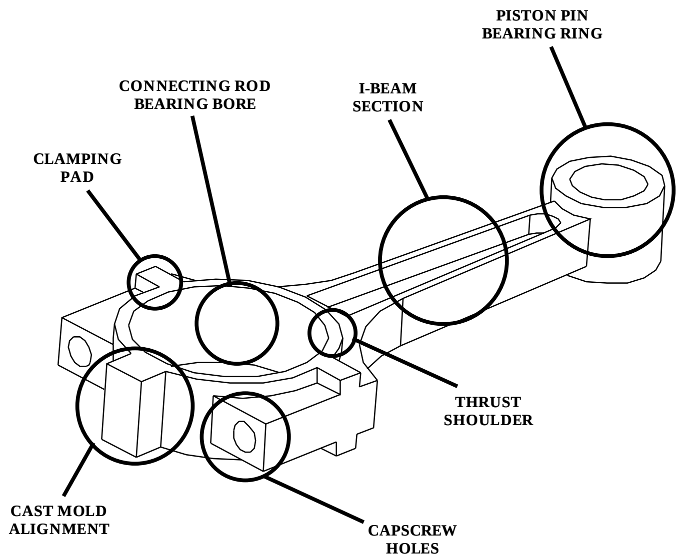


Figure 32 Features of design signficance on the connecting rod

Consider the functional significance of the areas highlighted in Figure 32. The functions illustrated are several of those necessary to aid the connecting rod in fulfilling its overall function, namely TRANSMITING FORCE from the piston chamber to the crankshaft. Examples of parts of the connecting rod associated with operational functions include “piston pin bearing ring,” “i-beam section,” and “thrust shoulder.” In addition, two areas of the part are highlighted that affect the manufacturing of the part, yet are design features specifically intended to do so. These features include “clamping pad,” and “cast mold alignment” in Figure 32.

Upon comparing Figure 32 with Figure 31, it is apparent that many of the features present in the set illustrated by Figure 31 map directly to the functional sections of the part shown in Figure 32. “Piston pin bearing ring” corresponds to feature 9, “i-beam section” corresponds (albeit indirectly) to features 5, 6, 7, 8, 10, 11 and 12, and “clamping pad” corresponds to features 2, 3, 13, and 14. Other features correspond to functional areas as well. Drawing the explicit links is left as an exercise for the reader.

3.2.3 Discussion of results

What is important to learn from this case study is that the VFR method is capable of recognizing features that have not only manufacturing significance, but design significance as well. This may seem to be a foregone conclusion, given that features themselves possess the ability to encapsulate design and/or manufacturing information. Given that, any “feature recognition” method ought to be capable of recognizing both types of features.

All too often, however, feature recognition methods are limited in their scope. Many “recognition” methods begin with a set of pre-defined manufacturing features and search only for those entities on the part that match its set of features. By definition, such methods cannot locate design features when such features differ significantly from manufacturing-type features. Only by defining features in a context-independent manner (via geometry and topology) can a method truly recognize both types of features.

Also of note is the fact that, once the VFR method recognizes the design features from the model and potentially links those features to other via interfeature relationships (e.g., FEATURE1 is planar with FEATURE2), an implicit constraint is made between those two features and the geometry that defines them. This is of particular importance in a variant design environment when designs may be incrementally modified from iteration to iteration. If implicit constraints exist between features, the system can tell a designer that "if you move this BOSS feature, not only will you impact the `clamping pad' functionality, but you will also affect the coplanar relationship the BOSS has with the RIB feature, which impacts the `assembly' functionality." Placing such information at the hands of designers makes a direct engineering type of system far closer to reality.

3.3 Case Study #3: Estimating Manufacturing Cost from Alternate Feature Sets

Oftentimes, it is not desirable (or possible) to generate single set of features for a given part volume and stock volume. Subtractive and additive features may interact with one another (as shown earlier in Figure **???**), suggesting several different ways in which the feature set can be interpreted from a manufacturing perspective. When this is the case, the question becomes: how should one feature set be judged relative to the others? This case study demonstrates a method by which such a determination is made.

Conducting manufacturability analysis using features is a popular course of research, but the field is hardly homogeneous. The flexibility of features enables a wide variety of different sorts of analysis. Research ranges from more focused methods to improve the design of sheet metal parts for manufacture [3], analyze the assemblability of a part [49], or reduce the set-up cost in process planning [7, 15], to more general approaches designed to analyze "manufacturability" [39, 23, 24].

For the purposes of this case study, we have chosen to measure manufacturability by a more quantifiable metric, namely manufacturing cost. In order to facilitate this analysis, we make use of a program called Cost Advantage[1]. The methodology by which Cost Advantage calculates manufacturing cost is discussed in Section 3.3.2.

In general the methodology used in this chapter is as follows:

I. generate the maximal simple feature set for a given part and given stock

II. generate alternate feature sets using the maximal simple feature set

III. for several feature sets, analyze its cost using Cost Advantage

IV. analyze the results and potentially change the design in an attempt to reduce manufacturing cost

V. begin the process anew

The analysis herein is done on an industrial base plate used in electronic assembly, shown in Figure 33. The remainder of the chapter focuses on generating alternate feature sets of the base plate, choosing several sets for cost analysis, and discussing what insights are gained from the cost analysis.

[1] "Cost Advantage" is a trademark of Cognition Corporation.