TECHNICAL RESEARCH REPORT

Feature Recognition for Manufacturability Analysis

by W.C. Regli, S.K. Gupta, D.S. Nau

T.R. 94-10

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Feature Recognition for Manufacturability Analysis[*]

Available as ISR TR94-10.

William C. Regli [†] Satyandra K. Gupta [‡] Dana S. Nau [§]

March 29, 1994

Abstract

While automated recognition of features has been attempted for a wide range of applications, no single existing approach possesses the functionality required to perform manufacturability analysis. In this paper, we present a methodology for taking a CAD model and extracting a set of machinable features suitable for generating all alternative interpretations of the model as collections of MRSEVs (Material Removal Shape Element Volumes, a STEP-based library of machining features). This set of MRSEVs is to be employed for manufacturability analysis. The algorithm handles a variety of features including those describing holes, pockets, slots, and chamfering and filleting operations. In addition, it considers elementary accessibility constraints for these features and is provably complete over a significant class of machinable parts the features describe. Further, the approach has low-order polynomial-time worst-case complexity.

1 Introduction

Automated recognition of features has been attempted for a wide range of applications, however no single existing approach is entirely suitable for use in manufacturability analysis. Although many of the CAD/CAM applications addressed in previous approaches have had compatible goals and functionality, it is often unclear what specific classes of objects, features, and feature interactions can be handled, making it difficult to evaluate their utility for manufacturability analysis.

We present in this paper a methodology for taking a CAD model and translating it into a set of features useful for performing manufacturability analysis in the domain of machined parts. We present algorithms capable of finding all the instances of a class of volumetric features corresponding to machining operations that occur in the alternative interpretations of the CAD model. Guaranteeing that all features from alternative interpretations are found is crucial to manufacturability analysis—where alternative interpretations are generated and evaluated in order to determine which one is optimal.

Although several approaches have previously been developed for recognizing features from CAD models, we address several issues that have not been adequately addressed by any single existing approach:

1. For purposes of integrating CAD with CAM, it is important to be able to get features that correspond directly to manufacturing operations—but such features are not often provided by existing approaches. Moreover, no standard schemes are used for representing features, therefore output of these systems cannot be directly used in downstream computer aided manufacturing applications.

   To address this problem, we use a class of features that are expressible as MRSEVs (Material Removal Shape Element Volumes) [15]. MRSEVs are volumetric features corresponding to machining operations on 3-axis milling machines. MRSEVs can be defined using EXPRESS (the official STEP information modeling language) and STEP form features. By employing a set of features based on a standard interchange format such as STEP, we have attempted to ensure that we are addressing a domain of machinable parts of interest to a large

---

[*] This work was supported in part by NSF Grants NSFD CDR-88003012, and DDM-9201779, and by the University General Research Board.

[†] Department of Computer Science and Institute for Systems Research, University of Maryland

[‡] Department of Mechanical Engineering and Institute for Systems Research, University of Maryland

[§] Department of Computer Science and Institute for Systems Research, University of Maryland

community. The authors believe this is the first feature recognition effort to address a feature class that has been defined using PDES/STEP.

2. Although many approaches have been developed for recognizing features from solid models of mechanical parts, the absence of a clear mathematical formalism for the problem has made it difficult to ensure completeness of these approaches. In particular, when features intersect with each other, this changes their topology and geometry in ways not taken into account by most existing feature recognition systems. Fast algorithms typically work only for limited classes of parts (e.g., $2\frac{1}{2}$-dimensional parts with no interactions among the features). Systems that attempt to handle more complex parts and feature interactions typically use algorithms that take exponential time in the worst case, or involve heuristic techniques that work in some cases but not others. Hence, it is often unclear what specific classes of parts and feature interactions can be handled by various existing approaches.

   To address this problem, we have developed an algorithm for recognizing any solid that can be described as the difference between a solid model (i.e., something describable with a finite boundary representation) of a piece of stock and an set of MRSEV instances. The subclass of the MRSEV library we recognize include hole, pocket, and edge-cut features, along with some elementary accessibility constraints for those features. The algorithm's time complexity is quadratic in the number of solid modeling operations.

3. In general, there may be several alternative interpretations of the design as different collections of machinable features, corresponding to different ways to machine the part. Determining which of these alternatives is most preferable requires considering the part dimensions, tolerances, and surface finishes, the availability and capabilities of machine tools and tooling, and fixturability constraints. However, most recognition systems do not present a systematic methodology that guarantees that all features of importance for the generation and evaluation of alternatives are found. Many try to generate a single interpretation for a given part—but in general, there may be several alternative interpretations of the part, each of which should be generated and examined to evaluate the manufacturability of the artifact.

   To address this problem, our feature recognition methodology is capable of finding all the features of importance to building the alternatives within a significant class of machinable parts. This is embodied in concept of completeness—i.e. finds *all* features occurring in the many alternate interpretations of the part regardless of whether they interact. We have shown previously that our approach is provably complete over the set of all solids in a mathematically specifiable class of parts described by a subclass of MRSEV features, even if the features intersect with each other in complex ways [23].

In our previous work [8, 23], we had focused on developing a formalization of the problem of recognizing machinable features expressible as MRSEVs and demonstrating provable completeness and complexity properties for our algorithms. This paper builds on these results, emphasizing the link between MRSEVs and machining operations, their relationship to evolving standards for data exchange, and the unique characteristics of our approach that will be required to perform manufacturability analysis.

This paper is organized as follows: Section 2 discusses related work. Section 3 presents our definitions. Section 4 presents our algorithm for recognizing MRSEVs. Section 5 presents potential applications of this feature recognition methodology for manufacturability analysis. Section 4.3 describes our current implementation. Section 6 discusses future extensions of our research and concluding remarks.

2 Related Work

Feature-based approaches have been very popular in a variety of CAD/CAM implementations, but different people have used the term to mean different things [28, 11, 7]. Significant amounts of work have been directed towards defining sets of form features to serve as a communication medium between design and manufacturing—but at present, most researchers are convinced that a single set of features cannot satisfy the requirements of both of these domains.

Most significant to the development of feature recognition systems for use in manufacturability analysis is recent work on feature recognition and generation of alternative feature interpretation for parts. For a comprehensive overview of feature-based manufacturing techniques, the reader is referred to [27].

Feature recognition has been considered an important research area in CAD/CAM integration and many different approaches have been developed over the last decade. The approaches of [4, 12] based on graph algorithms provide an excellent level of computational formality. However, while they have known algorithmic properties, they appear difficult to extend to realistic manufacturing problems. The grammatical methods of [19, 24] and some of the graph-based approaches are prone to combinatorial difficulties. The recent work in [6] describes promising techniques that combat the combinatorial problems by abstracting an approximation of the geometric and topological information in a solid model and finding features in the approximation. Corney and Clark [3] have had success extending the capabilities of graph-based algorithms to more general $2\frac{1}{2}$-dimensional parts.

Sakurai [26] emphasized the need to be able to extract some form of user defined feature types that may arise in specific applications and unites it with graph-based feature recognition. In more recent work, Sakurai and Chin [25] propose an algorithm for recognizing very general protrusions and cavities through "spatial decomposition and composition."

In one of the early efforts on feature extraction, Woo [32] proposed a method for finding general depression and protrusion features on a part through decomposing the convex hull of the solid model. The approach had several problems, including the existence of pathological cases in which the procedure would not converge. The non-convergence of Woo's approach has been solved in recent work by Kim [13]. Currently being addressed is how to extend this method from polyhedra to handle the general surfaces (for example, analytic surfaces such as cones and spheres) found in realistic parts.

The work of Henderson [10] was seminal in employing expert systems on the feature recognition problem. More recently, Henderson has employed graph-based methods and neural networks to recognize features [2, 20]. Kyprianou [16] presented the first effort to use grammars to parse solid models of parts for group coding.

The ability to handle interacting features has become an informal benchmark for feature recognition systems and has been the focus of numerous research efforts. The work of [5] included the formalization of a feature description language and employed frame-based reasoning algorithms to extract machining features for computer aided process planning. An aggressive approach to handle feature interactions and intersections was done by Marefat [18]. The work built on the representation scheme of [12] and used a novel combination of expert system and hypothesis testing techniques to extract surface features from polyhedral objects.

Perhaps the most comprehensive and formal approach to date for recognizing features and handling their interactions has been that of Vandenbrande [30]. It provides a computationally rigorous framework for recognizing a significant class of realistic machining features of interest for process planning via artificial intelligence techniques in combination with queries to a solid modeler. He formalized a set of feature classes and recognition "hints" for each class. The hints are extracted from the solid model and classified as to their potential for building a feature instance. A frame-based reasoning system then acts on the hints and attempts to complete a feature frame with information needed to make a maximal instance of a feature. [31] argues that the approach is complete over the set of all cases in which sufficient hints exist to identify all features that have to be found; however, no proof of this has been presented.

The recent work of [17] couples feature-based design and feature recognition to provide for incremental feature recognition. This type of approach recognizes changes in the geometric model as new or modified features while preserving the existing feature information. They also provide for some forms of customizability with use of a feature-definition language to add new features into the system.

3 Definitions and Notation

3.1 Basic Concepts

A *solid* is a manifold r-set with analytic bounding surfaces. If R is any solid, then $b(R)$ is the *boundary* of R, and $\iota(R)$ is the *interior* of R. Note that $R = \iota(R) \cup b(R)$ and that $\iota(R) \cap b(R) = \emptyset$. If R and R' are solids, then $R \cap^* R'$ is the *regularized intersection* of a and b, i.e., the closure of $\iota(R) \cap \iota(R')$. Similarly, $R \cup^* R'$ and $R -^* R'$ are the *regularized union* and *regularized difference*, respectively.

A *machined part* (or just a *part*) is the finished component to be produced as a result of a finite set of machining operations on a piece of *stock*, i.e., the raw material from which the part is to be machined. We will represent both the part and the stock as geometric solids. We use term *workpiece* to describe the state of stock after applying a subset of operation sequences. Throughout this paper, we let P be a solid representing a part, and S be a solid representing the stock from which P is to be made. The *delta volume* (i.e., the volume to be machined), is the solid $\Delta = S -^* P$.

3.2 Material Removal Shape Element Volumes (MRSEVs)

3.2.1 PDES/STEP

STEP is the International *St*andard for the *E*xchange of *P*roduct Model Data being developed by the International Organization for Standardization (ISO). PDES stands for *P*roduct *D*ata *E*xchange using *S*TEP and it represents the activity of several organizations in the United States in support of STEP. The organizations involved with PDES comprise many corporate, government, and standards development entities.

Describing data in STEP is handled by defining an information model in the EXPRESS data modeling language [29] for each type of data required. Once an information model is defined, data for representing a specific product can be represented by using the STEP rules for mapping EXPRESS to a physical file [1]. The EXPRESS model defines the data entities that describe the class of objects in the domain.

3.2.2 The MRSEV Hierarchy

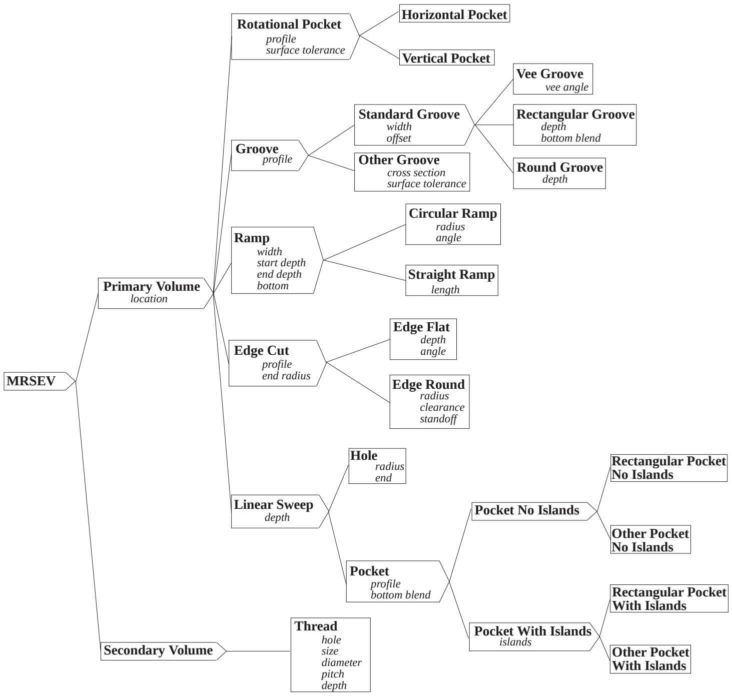
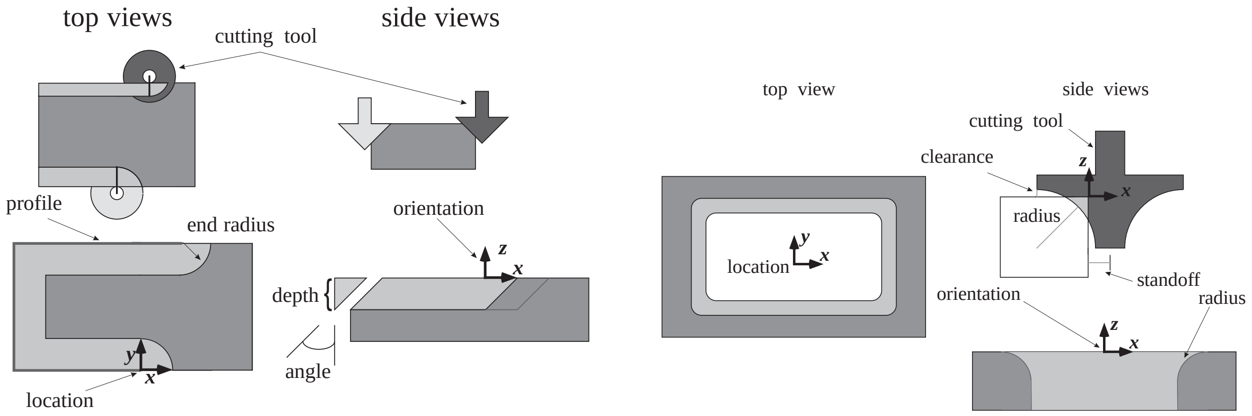


Figure 1: The features in the hierarchy of MRSEVs and their attributes.

Kramer [15] developed a library of material removal shape element volumes (MRSEVs) as a means of categorizing the shapes of volumes to be removed by machining operations on a 3-axis machining center, such as drilling and milling. MRSEVs can be defined using the EXPRESS modeling language and STEP form features. Kramer [15] has written such definitions for a subset of the MRSEV library, and has defined the rest of the MRSEV library using an EXPRESS-like language.

The MRSEV hierarchy provides a framework for describing a large class of volumetric entities of interest to machining. Each entity type has a number of required and optional attributes. MRSEV instances have been used for applications such as process planning and NC-program generation [14]. Kramer's primary MRSEV types include linear swept features, edge-cut features, ramps, general grooves, and rotational pockets.



Attribute of edge-flat	**Type**
location	point
orientation	vector
profile	edge loop
depth	real number
angle	real number
end radius	optional real number

Attribute of edge-round	**Type**
location	point
orientation	vector
radius	real number
profile	edge loop
clearance	real number
standoff	real number
end radius	optional real number

(a) (b)

Figure 2: Specification for the two subclasses of edge-cut MRSEV features.

MRSEVs are geometrically and topologically defined volumetric features. Information about design attributes such as tolerances, surface finishes, or available machine tools that would yield specific operations to machine the volumes are not part of their definitions. While the CAD models we consider have attributes denoting tolerances and other machining constraints, selection of the appropriate operations is outside the specifications for our feature recognition system. In the context of this approach, consideration of these types of machining constraints and choices for specific operations that machine a MRSEV volume is performed during the manufacturability analysis, as will be discussed in Section 5. The operations used to machine a MRSEV will depend on the cost and availability of tools and machines to satisfy these design attributes and the parameters considered to analyze manufacturability.

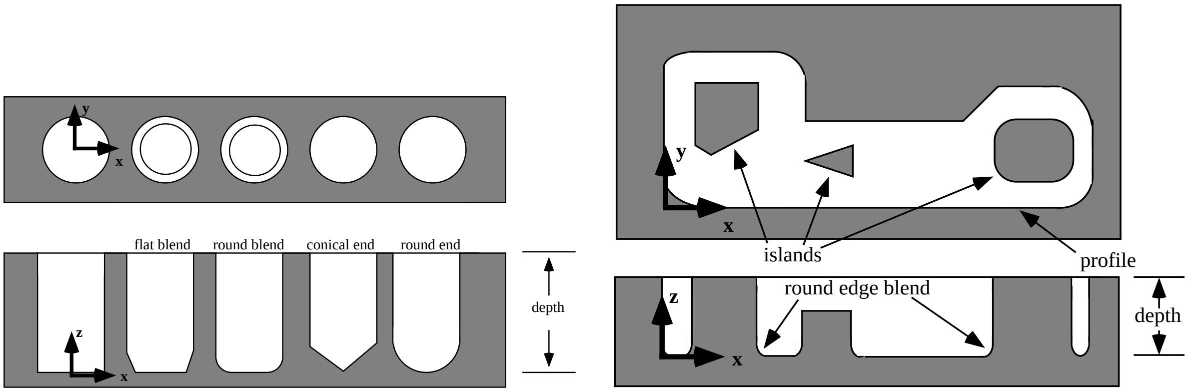
For the purpose of this paper we confine our domain to the edge-cut and linear swept features, i.e., holes, pockets, and pockets with islands, chamfers, fillets, countersinks, and edge blends. Kramer defines linear swept feature as a shape resulting from sweeping a closed profile of edges along a straight line perpendicular to the plane of the profile[1]. An edge-cut feature results from sweeping the flat or round edge of an angled tool along a, possibly open, profile of edges. The product of this kind of MRSEV feature is typically a flattened or rounded edge on the part, such as a chamfer. For manufacturability analysis, we have added criteria (such as accessibility and existence of corner radii for convex pocket corners) to MRSEVs to ensure the volumes recognized are in some way machinable. Figs. 2(a) and 2(b) present illustrations for two the two classes of edge-cut MRSEV features: edge flats and edge rounds. Illustrations of MRSEV holes and pockets can be found in Figs. 3(a) and 3(b).

3.2.3 MRSEV Instances

The MRSEVs are parameterized solids. A specific instance of one of these MRSEVs can be instantiated by

---

[1] In the case of a pocket with islands, an island is considered to be a subfeature defined by its own closed profile.



Attribute of hole	**Type**
location	point
orientation	vector
depth	positive real number
radius	positive real number
end	end condition

Attribute of pocket	**Type**
location	point
orientation	vector
depth	positive real number
profile	edge loop
bottom blend	edge blend
islands	set of one or more islands
Attribute of island	**Type**
profile	edge loop
bottom blend	optional edge blend
height	positive real number

(a) (b)

Figure 3: MRSEV Holes and MRSEV Pockets with islands.

assigning specific choice of attribute values. For example, suppose we choose the following attribute values:

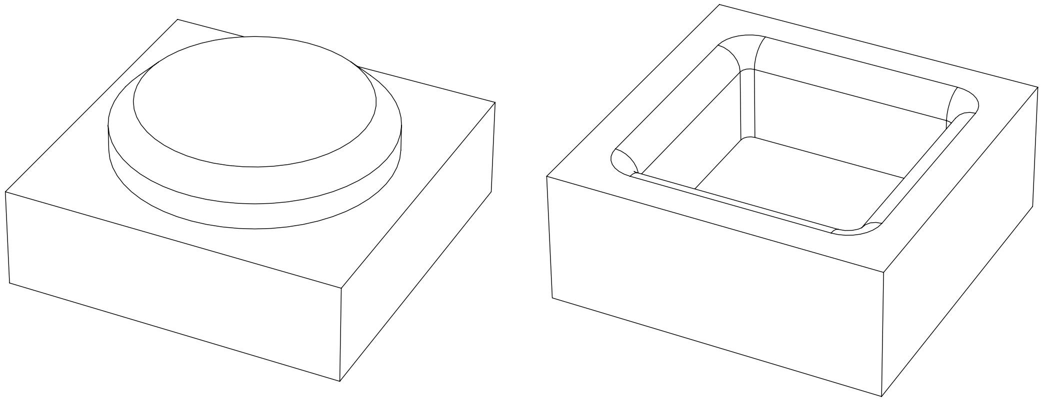
location = $(0, 10, 4)$;
orientation = $(0, 0, 1)$;
profile = $\{e_1\}$;
depth = 7;
angle = 45 degrees;
endradius = none;

This would define the chamfer illustrated in Figure4 (a). Similarly, the following values would define a MRSEV edge-round pictured in Figure4 (b):

location = $(0, 0, 30)$;
orientation = $(0, 0, 1)$;
radius = 7;
profile = $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$;
clearance = 0;
standoff = 0;
endradius = *none*;

3.3 Correspondence Between Machining Operations and MRSEVs

To perform a machining operation, one starts out with a rotating cutting tool. The cutting tool is mounted on a large machine tool, and the total volume occupied by the cutting tool and the machine tool is quite large. But we will only be interested in some small portion of this total volume, namely the portion that actually gets close to the workpiece. We will call this portion the *tool volume*, and we will denote it by T. The boundary of T consists of both cutting and non-cutting surfaces. To perform the machining operation, one sweeps the tool volume T along some



(a)
Edge-flat MRSEV

(b)
Edge-round MRSEV

Figure 4: Instances of edge-cut MRSEVs.

trajectory. Informally, the solid consisting of the set of points hit by the cutting surfaces of T as it is swept along the trajectory will be the material removed by a machining operation.

MRSEVs can be used to represent volumes which can be removed during machining. For example, a MRSEV hole represents a volume which can be removed by a drilling operation, and a MRSEV pocket represents a volume which can be removed by an end or face milling operation. It is worth noting that the "pocket" MRSEV is used not only to represent what is usually called a pocket, but also to represent a large variety of milled shapes such as steps, profiles, slabs, etc.

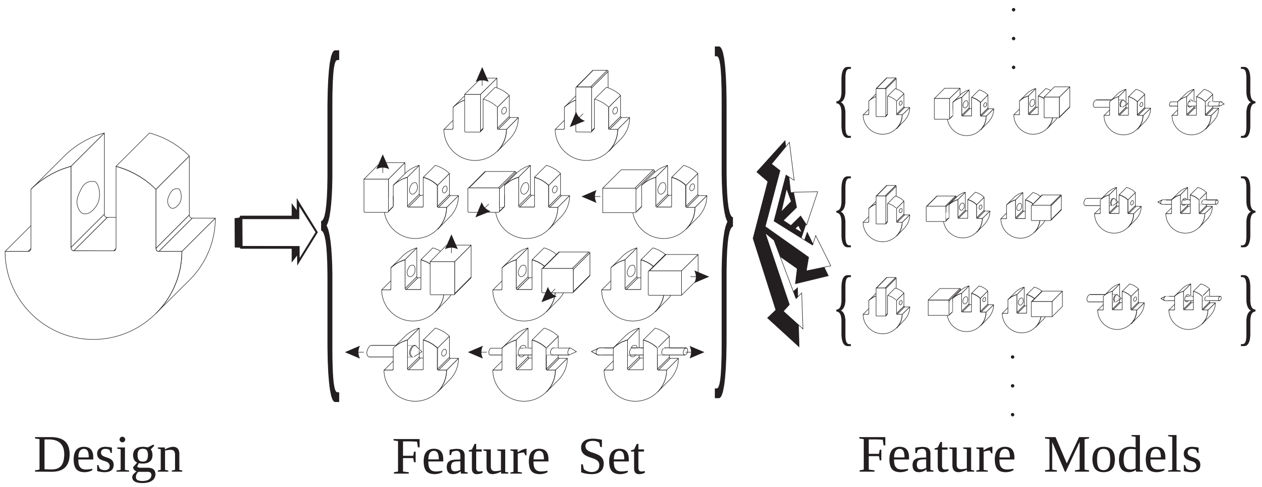


Figure 5: Examples of feature models.

4 Recognizing MRSEVs

Given solids representing the part P and the stock S, we are interested in finding the set of all MRSEVs that correspond to useful machining operations that can be used to create P. In this section we present our methodology for recognizing MRSEVs.

The input to the feature recognition system is a pair of solids representing the initial stock S, and the final part P. Our objective is to have the system produce a set of MRSEV feature instances that can be used to generate and evaluate alternative interpretations of P. A *feature model* represents an interpretation of the part as a set of MRSEV features.

Feature Models. Let P be the given part and S be the given stock. We define a *feature model* of P and S to be a finite set of MRSEV instances M having the following properties:

1. If we subtract the MRSEVs in M from S, we get P; i.e., $S -^* \cup^*_{m \in M} m = P$.
2. No MRSEV in M is redundant, i.e., for every MRSEV $l \in M$, $S - \cup^*_{m \in M - \{l\}} m \neq P$.

Intuitively, a feature model is an interpretation of the delta volume as a set of machining features. For example, the set $\{h, s1, s2, s3\}$ shown in Figure 5 is a Feature Model.

Given a realistic part, there will be a large (theoretically speaking, infinite) number of MRSEVs that may be used in a model of the part. We will only be interested in recognizing those MRSEVs which will be useful in CAM applications. Therefore, we define following two restrictions on MRSEVs we will consider:

Valid MRSEVs. A MRSEV m is *valid* for a given part P, if:

1. m creates some portion of the boundary of P (i.e., $b(m) \cap^* b(P) \neq \emptyset$).
2. m does not intersect P (i.e., $m \cap^* P = \emptyset$).

Primary MRSEVs. A *primary* MRSEV for a part P and stock S is any valid MRSEV m, that satisfies the following conditions:

1. Every valid MRSEV n that contains m (and has the same location and orientation) also has the same effective volume as m (i.e., if $m \subset n$ then $\text{eff}(n, S) = \text{eff}(m, S)$).
2. Every valid MRSEV n that is contained in m (and has the same location and orientation) has a smaller effective volume (i.e., if $n \subset m$ then $\text{eff}(n, S) \subset \text{eff}(m, S)$).

The output of the feature recognizer is a primary *Feature Set*, $\mathcal{M}$. This is a finite set whose elements are all primary MRSEV instances from some feature model.

An arbitrary part and stock may still present a problem because they still could contain an infinite number of instances of primary MRSEVs, giving rise to an infinitely large feature set. In order to address this, we consider parts which have feature models containing primary MRSEVs with the following characteristics:

1. For any hole MRSEV in a model, the delta volume contains either a subface of its cylindrical face or its entire ending surfaces;
2. For any pocket MRSEV in a model, either a subface of its bottom face is present in the delta volume, or else it is a through pocket with one non-planar or at least two of its non-parallel planar side faces present in delta volume.
3. For any edge-cut MRSEV in a model, the complete surface produced by the cutting edge of the tool is present in the delta volume.

For any part satisfying these restrictions, there will be only a polynomial number of primary MRSEV instances [23].

A MRSEV recognition algorithm is *complete* if it returns the set of all primary MRSEV instances that appear in any of the feature models of P and S. As discussed in the next section, the above restrictions provide sufficient information to be able to identify all instances of hole, pocket, and edge-cut MRSEVs, and thus they allow us to develop a MRSEV recognition algorithm that is complete over all parts satisfying these restrictions. The proof of completeness for the cases of hole and pocket MRSEVs was given previously in [23].

4.1 MRSEV Recognition Algorithms

To extract instances of MRSEVs, we start with a solid model of the part, and obtain data about it via queries to the solid modeling system. One of the most popular approaches for representing geometric solids is the boundary-representation approach, in which the solid is represented in terms of the faces that bound it. Our algorithm uses solid modeling operations basic to boundary-representation solid modeling systems. Proceeding from the observation that every face of the delta volume must be on the surface of some MRSEV instance m, the algorithm constructs all primary MRSEV instances that contain m. It does this by traversing the faces of the delta volume and instantiating the primary MRSEVs capable of covering all or a portion of each face. A high-level description of the MRSEV recognition algorithm is given in Fig.6.

```
Algorithm 4.1 Recognize_MRSEVs($P, S$)
INPUT: solid models of a part $P$ and stock $S$
OUTPUT: a primary feature set, $\mathcal{M}$. Initially, $\mathcal{M} = \emptyset$.
        for each face $f$ of $S -^* P$ do
            if surface type of $f$ == PLANE
                $\mathcal{M} = \mathcal{M} \bigcup$
                        Find_Holes_from_Planar_Face($f, P, S$) $\bigcup$
                        Find_Pockets_from_Planar_Face($f, P, S$)
                        Find_Edge-Flats_from_Planar_Face($f, P, S$)
                        Find_Edge-Rounds_from_Planar_Face($f, P, S$)
            if surface type of $f$ == CYLINDER
                $\mathcal{M} = \mathcal{M} \bigcup$
                        Find_Holes_from_Cyl_Face($f, P, S$) $\bigcup$
                        Find_Pockets_from_Cyl_Face($f, P, S$)
                        Find_Edge-Flats_from_Cyl_Face($f, P, S$)
                        Find_Edge-Rounds_from_Cyl_Face($f, P, S$)
            if surface type of $f$ == CONE
                $\mathcal{M} = \mathcal{M} \bigcup$
                        Find_Holes_from_Conical_Face($f, P, S$) $\bigcup$
                        Find_Pockets_from_Conical_Face($f, P, S$)
                        Find_Edge-Flats_from_Conical_Face($f, P, S$)
                        Find_Edge-Rounds_from_Conical_Face($f, P, S$)
            if surface type of $f$ == TORUS
                $\mathcal{M} = \mathcal{M} \bigcup$
                        Find_Holes_from_Toroid_Face($f, P, S$) $\bigcup$
                        Find_Pockets_from_Toroid_Face($f, P, S$)
                        Find_Edge-Flats_from_Toroid_Face($f, P, S$)
                        Find_Edge-Rounds_from_Toroid_Face($f, P, S$)
            if surface type of $f$ == SPHERE
                $\mathcal{M} = \mathcal{M} \bigcup$
                        Find_Holes_from_Spherical_Face($f, P, S$) $\bigcup$
                        Find_Pockets_from_Spherical_Face($f, P, S$)
                        Find_Edge-Flats_from_Spherical_Face($f, P, S$)
                        Find_Edge-Rounds_from_Spherical_Face($f, P, S$)
        end for
        delete all flat bottomed holes from $\mathcal{M}$// subsumed by pockets
        return($\mathcal{M}$)
```

Figure 6: A high-level description of a MRSEV recognition algorithm.

A MRSEV instance is a parameterized solid. For each type of MRSEV, we have a procedure that constructs an instance of a MRSEV with the given parameters as attribute values:

New_Hole(**location,orientation,depth,radius,end**)
New_Pocket(**location,orientation,depth,profile,islands**)
New_Edge-Flat(**location,orientation,depth,radius,end**)
New_Edge-Round(**location,orientation,depth,radius,end**)

```
Algorithm 4.2 FIND_HOLES_FROM_CYL_FACE($f, P, S$)
INPUT: solid models of a part $P$ and stock $S$, a cylindrical face $f$.
OUTPUT: a primary feature set, $\mathcal{M}'$, containing all holes that can be found from $f$.
        radius = radius of $f$
        orientation = axis of $f$
           // $e$ is a face
        FIND_HOLE_END(orientation, $e$)
           // on exit from FIND_HOLE_END, orientation
           // points toward the hole end, if there is one
           // $e$ is the hole end surface
        if there is a hole end $e$ then
           end = surface type of $e$
           location = position of $e$
           depth = distance from location to outside stock on orientation
           $m$ = NEW_HOLE(location,orientation,depth,radius,end)
           if ($m \cap^* P == \emptyset$) then
              $\mathcal{M}' = \{m\}$
        else
           location = position outside stock on orientation
           depth = distance from location to outside stock on orientation
           end = conical_end
           $m_1$ = NEW_HOLE(location,orientation,depth,radius,end)
           location = position outside stock on -orientation
           $m_2$ = NEW_HOLE(location,-orientation,depth,radius,end)
           if ($m_1 \cap^* P == \emptyset$) then
              $\mathcal{M}' = \{m_1\}$
           if ($m_2 \cap^* P == \emptyset$) then
              $\mathcal{M}' = \{m_2\}$
        return($\mathcal{M}'$)
```

Figure 7: Algorithm for recognizing MRSEV holes from cylindrical part surfaces.

The boundary of a MRSEV instance is made up of different types of faces. Each kind of face (planar face, conical face, etc.) may be part of the boundary of one or more types of MRSEVs. For example, a cylindrical surface could be considered as the side face of a MRSEV hole, as a corner radius of a MRSEV pocket, or the surface produced by an edge-round MRSEV. Constructing a set of primary MRSEV instances covering a particular face depends on the type of face we are building from. In the case of a cylindrical face, we want to try to instantiate each type of primary MRSEV feature that might have produced it : a MRSEV hole, an MRSEV edge-round, and one or more primary MRSEV pockets.

To accomplish this, for each type of face we have functions that, via queries to the solid modeler, return the set of primary MRSEVs that can contain all or part of that face. For each type of MRSEV, the function finds values for the attributes of primary MRSEV instances of that type and constructs each. Hence, by considering every face in the delta volume (i.e. every face that needs to be machined) the set $\mathcal{M}$ of all of primary MRSEV instances can be built.

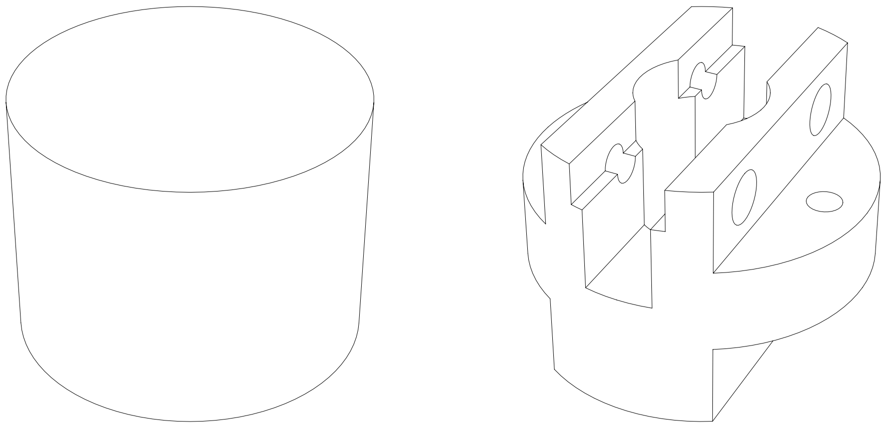
For example, we have functions for building MRSEVs from cylindrical faces:

FIND_HOLES_FROM_CYL_FACE(f, P, S)
FIND_POCKETS_FROM_CYL_FACE(f, P, S)
FIND_EDGE-FLAT_FROM_CYL_FACE(f, P, S)
FIND_EDGE-ROUND_FROM_CYL_FACE(f, P, S)

Each of these functions return the set of instances of primary MRSEVs that can be found from a given cylindrical face f. Similarly, for each type of face and type of MRSEV there is a function

```
Algorithm 4.3 Find_Pockets_from_Planar_Face($f, P, S$)
INPUT: solid models of a part $P$ and stock $S$
OUTPUT: a primary feature set, $\mathcal{M}'$, containing all pockets that can be found from $f$.
    // the case for a non-through pocket; i.e. $f$ is part of a bottom face
    orientation = normal vector of $f$
    intersect $P$ with half-space above the plane of $f$
    sweep the intersected $P$ on orientation for depth distance
    profile = slice the swept $P$ on orientation through its middle
        // the slice would be at a point on orientation and distance depth/2
        // from location; there may be several profiles found by
        // the slice, we are interested in the one that intersects the line through
        // location on orientation
    islands = inner edge loops of profile
    location = position of profile
        // choice of this will depend on the application, for example it could be simply
        // the location of one of the vertices of one of profile's bounding edges
    depth = distance from location to outside stock on orientation
    m = New_Pocket(location,orientation,depth,profile,islands)
    if ($m \cap^* P == \emptyset$) then
        $\mathcal{M}' = \mathcal{M}' \bigcup \{m\}$
    // the case for a through pocket
    for each face $f'$ of $S -^* P$, $f' \neq f$, not $f' \,||\, f$ do
        orientation = cross product of the vectors normal to $f$ and $f'$
        sweep $P$ on orientation for depth distance
        profile = slice the swept $P$ on orientation through its middle
            // same as above
        islands = $\emptyset$
            // no islands, as it is a "through" pocket
        location = position of profile
            // again, choice of this will depend on the application
            // for example, the location of one of the vertices of one of
            // profile's bounding edges
        depth = distance from location to outside stock on orientation
        $m_1$ = New_Pocket(location,orientation,depth,profile,islands)
        $m_2$ = New_Pocket(location,-orientation,depth,profile,islands)
        Offset_Pocket($m_1$)
        Offset_Pocket($m_2$)
        if ($m_1 \cap^* P == \emptyset$) then
            $\mathcal{M}' = \mathcal{M}' \bigcup \{m_1\}$
        if ($m_2 \cap^* P == \emptyset$) then
            $\mathcal{M}' = \mathcal{M}' \bigcup \{m_2\}$
    return($\mathcal{M}'$)
```

Figure 8: Algorithm for recognizing MRSEV pockets from planar part surfaces.



(a): stock (before machining) (b): part (after machining)

Figure 9: An example part and stock.

Find_Holes_from_Planar_Face(f, P, S)
Find_Pockets_from_Planar_Face(f, P, S)
Find_Edge-Flats_from_Planar_Face(f, P, S)
Find_Edge-Rounds_from_Planar_Face(f, P, S)
Find_Holes_from_Conical_Face(f, P, S)
Find_Pockets_from_Conical_Face(f, P, S)
Find_Edge-Flats_from_Conical_Face(f, P, S)
Find_Edge-Rounds_from_Conical_Face(f, P, S)
etc...

Space does not permit elaboration on each of the functions for constructing primary MRSEVs from each kind of face. We present below two of the more interesting ones: Find_Holes_from_Cyl_Face and Find_Pockets_from_Planar_Face. We present pseudo-code outlines of these algorithms, for greater detail the reader is referred to [23]. Implementation of these will vary depending on the functionality of the modeling system chosen. The outline for algorithm 4.2, for builing instances of MRSEV holes, is in Fig.7.

Two situations need to be considered when building a MRSEV pocket from a planar face f: the first is where f contains part of the bottom face of a MRSEV pocket; the second is where f contains part of a side face of the pocket. In the first case, we obtain the profile of a primary MRSEV pocket by sweeping the part of the solid P lying above the face f along the orientation of the pocket and taking a cross-section of the resulting volume. This guarantees we will obtain a maximal profile that does not intrude into the part. For the second case, we are dealing with only through pockets. Hence, for each possible orientation for a through pocket we consider a cross-section of the part P after it has been swept in that direction. Algorithm 4.3 for constructing MRSEV pocket instances is in Fig.8.

Example. Figure 9 shows an example part. Let us assume that this part will be machined from a rectangular stock. For this part, Figure 10 shows the various MRSEVs identified by our algorithm. In this case, feature set is

$$\mathcal{M} = \{h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, h11, h12, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10\}.$$

4.2 Feature Offsetting

In many cases, the most cost effective way to machine a MRSEV involves performing the machining operation using the largest possible tool. Such situations may require that the tool moves outside the boundary of the stock material. For a MRSEV pocket, the edge profile denoting its outline may present unnecessarily strict constraints on the selection of tooling and operations. To take this into account, when the edges of the MRSEV pocket profile do not all lie on the part boundary we perform feature offsetting.

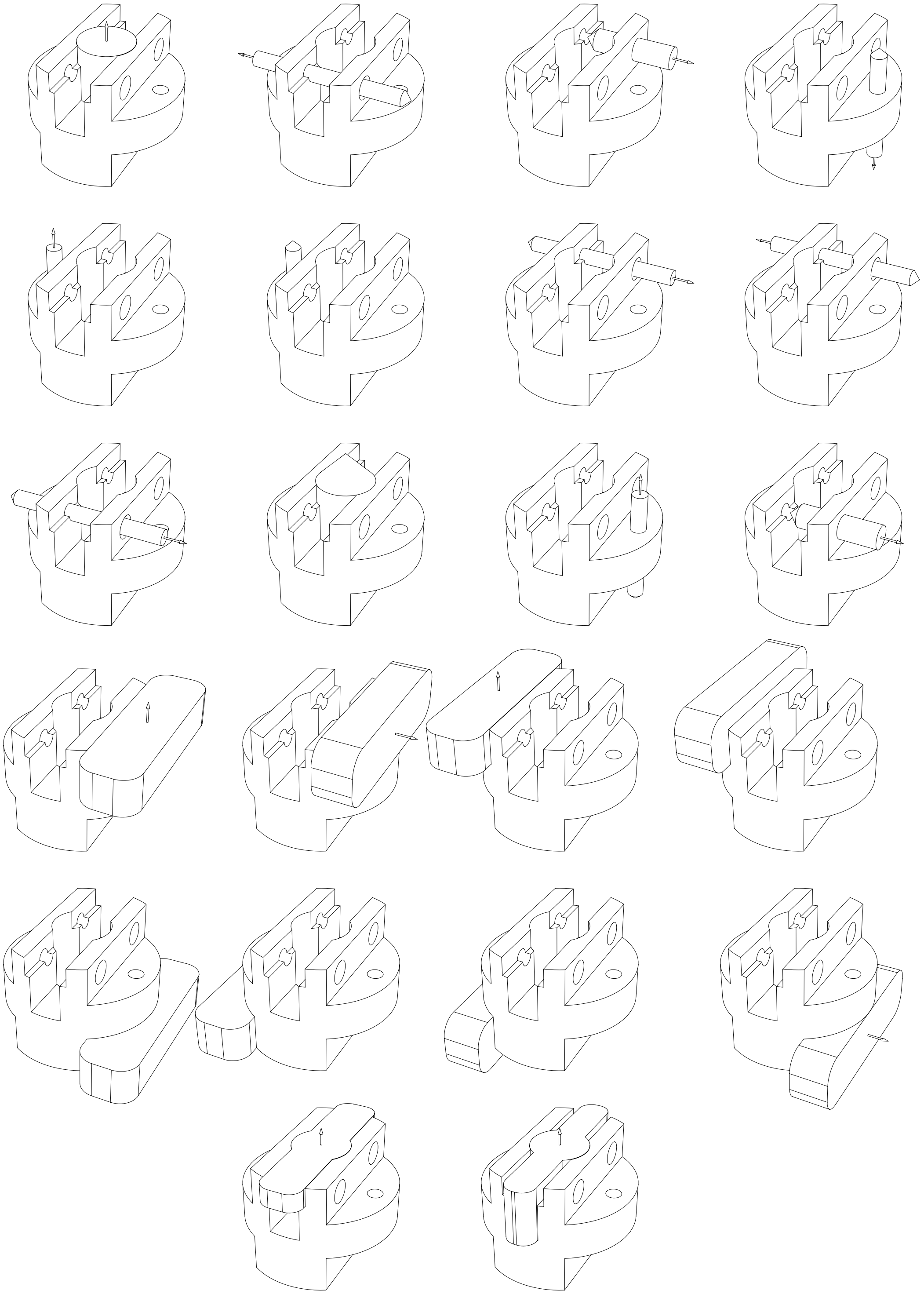


Figure 10: MRSEVs identified by our algorithm for the part shown in Figure 9.

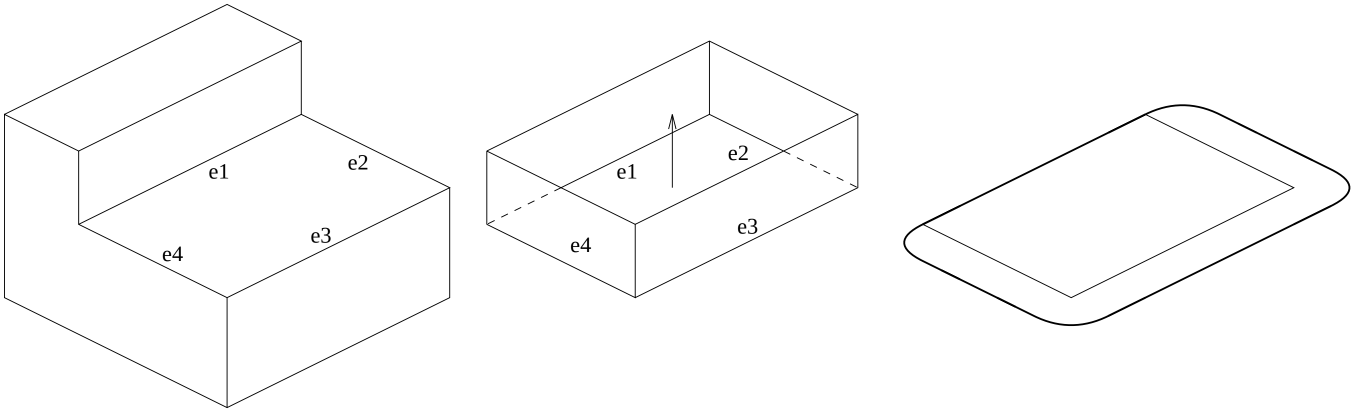


Figure 11: An example of feature offsetting.

Given an instance of a pocket MRSEV, the profile corresponding to its bottom face contains two types of edges: (1) *closed edges:* those which belong to pocket faces whose exterior's (with respect to the volume of the pocket) are incident with some portion of the surface of the part and (2) *open edges:* those belonging to pocket faces whose exterior's are incident exclusively with a portion of the surface of the stock.

After finding a MRSEV pocket profile containing open edges, the profile is offset to take into account the radius of the largest appropriate tool that may be used to machine the volume. This value can be thought of as a parameter to the feature recognizer. During offsetting, the open edges of a MRSEV pocket are found are offset by the tool radius in order to allow the tool to move on or outside of these edges during machining. This modifies the pocket volume to make it more appropriate for machining.

Figure 11 provides an illustration of the feature offsetting process. In the figure, edges e_2, e_3, e_4 have been offset to take into account the radius of a machine tool.

4.3 Implementation

We have built a proof-of-concept implementation of this feature recognition methodology in C++ using version 3.0.1 of the AT&T C++ compiler from SUN Microsystems, version 1.4.1 of Spatial Technologies' ACIS© solid modeling kernel, and the NIHCL C++ Class Library developed at the National Institutes of Health. Also being employed in our development efforts are Ithaca Software's HOOPS© Graphics System and the Tcl/Tk embeddable command language and user interface toolkit developed at the University of California at Berkeley.

The current implementation of the MRSEV recognizer conforms to the framework outlined in [22, 23] with the exception of bottom blended pockets and some cases of through pockets and the addition of the edge-cut features and feature offsetting presented in this paper. We have omitted bottom blends on pockets in the current implementation because they are not crucial to the downstream application of manufacturability analysis. Implementation for general through pockets was restricted by the current version of the ACIS© application procedural interface. At the time of this writing, we are extending the ACIS© application procedural interface to provide all the functionality described in [22, 8, 23].

4.4 Important Characteristics of the Algorithm

Our MRSEV recognition methodology has following distinguishing characteristics:

1. *Interface with standards:* It recognizes features from the MRSEV feature library. The authors believe this is the first feature recognition effort to address a feature class that has been defined using PDES/STEP.

2. *Features correspond to machining operations:* The MRSEV features addressed correspond directly to general machining operations on 3-axis machining centers.

3. *Feature Offsetting:* When possible, the MRSEV features recognized are offset to account for the dimensions of the milling tool to be used. The new offset features correspond more naturally to the area which will be

machined to produce the desired removal volume. This is significant to the our application of manufacturability analysis in that it simplifies costly and complex tool sweep profiles that might be produced by constructing features solely from the part and the delta volume.

4. *Accessibility:* The formulation of feature accessibility can include considerations about the possible physical mountings for the tool. This represents an improvement over methodologies that consider only the semi-infinite accessibility of feature surfaces.

5. *Ability to guarantee completeness over a significant class of parts:* Our approach encompasses many parts of direct interest to machining and manufacturability analysis application.

6. *Polynomial-time worst-case complexity:* This represents an improvement over the potentially exponential cost of subgraph matching and rule-based systems. If each solid modeling operation took unit time, the version of the algorithm we present here would run in time $O(n^2)$.[2]

   There are no definitive complexity results for the operations common to solid modeling systems. The actual cost will depend on the particular data structure chosen and its implementation—but it is believed that algorithms for computing booleans, for example, take time somewhere between $O(n^2)$ and $O(n^4)$. In this case, the total run time of our algorithm would be between $O(n^4)$ and $O(n^6)$.

5 Generating and Evaluating Feature Models from a Feature Set

Since the main focus if this paper has been to present a methodology for finding all features that are elements of feature models, we will not discuss the specifics of these applications. Rather our goal in this section is to outline the potential CAM applications that exploit the properties of this feature recognition methodology. For more information on the specifics of these applications, readers are referred to [8, 9].

After finding the set $\mathcal{M}$, the next step is to use these features to generate feature models for the design. Many times, the set $\mathcal{M}$ of all primary MRSEVs contains redundant MRSEVs (i.e., same portion of the delta volume is covered by more than one MRSEV). For most CAM applications, we will be interested in collections of MRSEVs which do not have any redundant elements (i.e., we don't want to machine the same volume twice). As defined in Section 4, MRSEV models are collections of MRSEVs which are sufficient for machining a given part and do not include any redundant elements. In general, for a given part there may be more than one feature model, each one corresponding to a potential way of making the part. Since each feature model is basically an irredundant set cover for the set $\mathcal{M}$, models can be generated using variations on irredundant-set-covering techniques[21], and use pruning heuristics to discard unpromising models.

Consider an evaluation function which estimates the manufacturability of a given feature model. In most of the cases, we are interested in finding the feature model which optimizes this evaluation function. For example, an evaluation function might use the feature model to consider production cost, production time or other factors related to manufacturability. In order to guarantee that the solution found is indeed optimal, it becomes important for the feature recognition procedure to be complete over the class of parts being considered. In practice this may not always be feasible, however analysis of completeness will produce information useful for determining the limitations of the system and identifying the potential sources for problems when they do occur.

Besides optimizing the evaluation function value, a feature model may need to satisfy additional constraints. For example, in case of process planning, operations associated with the feature model should be capable of meeting the tolerance requirements. Moreover, for a MRSEV model to be useful for process planning, there must exist a sequence of machining operations such that during all stages of machining, the intermediate workpiece geometry is suitable for fixturing and setup. Given a candidate operation sequence, the machining data for that sequence, the MRSEV's dimensions, and the material from which the part is to be made, we can evaluate whether or not it can satisfactorily achieve the tolerance specifications. As there may be many ways to achieve these constraints, it is important that the feature recognizer produce a satisfactory set for evaluation of all of these alternatives.

---

[2] We assume the solid S is represented by some boundary data structure, in general, the size n_S of this data structure will be $n_S = O(E_S)$ where E_S is the number of edges of the solid. For the worst case of these data structures, we can say this size is $O(n_S)$ where $n_S = E_S + V_S + F_S$ and E_S, V_S, and F_S are the number of edges, vertices, and faces of S respectively. Euler's equation tells $2 = V - E + F$, hence we can simplify this to be $n_S = 2 + 2E$ or just $n_S = O(E_S)$.

6 Conclusions

We have described our approach for recognition of machining features from CAD models for the purposes of manufacturability analysis. The algorithms we present take a CAD model and extract all instances of MRSEV features occurring in any of the alternative feature models for the given part. We have proven the approach to be complete over a significant class of parts.

Some of the primary characteristics of our approach are as follows:

1. While various CAD and CAM applications may have compatible goals and functionality, their specific details have been different enough that integration has proven difficult. To address this, our approach recognizes features from the MRSEV library, offering the possibility of compliance with the well known STEP standard. The authors believe this is the first feature recognition effort to addresses PDES/STEP definable feature classes.
2. In addition to feature recognition, these algorithms can be viewed as a means of translation from a solid model to a STEP representation. This is of potential significance for data exchange applications.
3. Our approach handles a variety of hole, pocket, and edge-cut MRSEVs, along with accessibility constraints for those features.
4. The algorithm's time complexity is quadratic in the number of solid modeling operations.
5. The feature recognition algorithm is provably complete over a significant class of realistic parts [23], even if the features intersect with each other in complex ways. This property is important for tasks such as manufacturability analysis, in which determining the manufacturability of a design may require trying a number of alternative possibilities to see which one is best.

Near-term goals include incorporating a more sophisticated definition of accessibility and continuing to enhance our implementation. Medium-term directions include extending our results and procedures to include other MRSEVs and exploring techniques for the simplifying the model in order to achieve a reduction in complexity (as done in [6]).

Further, we hope to exploit the object-oriented structure to the MRSEV hierarchy to support extensibility and user-defined features. We are currently considering how to use an object-oriented paradigm to generalize the results and algorithms to other feature hierarchies and user-defined feature classes.

References

[1] J. Altemueller. Mapping from express to physical file structure. Technical Report ISO TC184/SC4 Document N280, International Organization for Standardization, 1988. September.

[2] S. H. Chuang and M. R. Henderson. Three-dimensional shape pattern recognition using vertex classification and the vertex-edge graph. *Computer Aided Design*, 22(6):377–387, June 1990.

[3] J. Corney and D. E. R. Clark. Method for finding holes and pockets that connect multiple faces in $2\frac{1}{2}$d objects. *Computer Aided Design*, 23(10):658–668, December 1991.

[4] Leila De Floriani. Feature extraction from boundary models of three-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8), August 1989.

[5] Xin Dong. *Geometric Feature Extraction for Computer-Aided Process Planning.* PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, USA, 1988.

[6] R. Gadh and F. B. Prinz. Recognition of geometric forms using the differential depth filter. *Computer Aided Design*, 24(11):583–598, November 1992.

[7] N.N.Z. Gindy. A hierarchical structure for form features. *Int. J. Prod. Res.*, 27(12):2089–2103, 1989.

[8] Satyandra K. Gupta, Thomas R. Kramer, Dana S. Nau, William C. Regli, and Guangming Zhang. Building MRSEV models for CAM applications. Technical Report ISR TR93-84, The University of Maryland, December 1993. Accepted for publication in *Advances in Engineering Software*.

[9] S.K. Gupta and D.S. Nau. A systematic approach for analyzing the manufacturability of machined parts. Technical Report TR 93-76, Institute for Systems Research, University of Maryland, College Park, Md-20742, 1993.

[10] Mark R. Henderson. *Extraction of Feature Information from Three-Dimensional CAD Data.* PhD thesis, Purdue University, West Lafayette, IN, USA, 1984.

[11] K. E. Hummel. The role of features in computer-aided process planning. In *Proceedings of Feature Symposium*, Woburn, Boston, MA, August 1990.

[12] S. Joshi and T. C. Chang. Graph-based heurisitcs for recognition of machined features from a 3D solid model. *Computer-Aided Design*, 20(2):58–66, March 1988.

[13] Y. S. Kim. Recognition of form features using convex decomposition. *Computer Aided Design*, 24(9):461–476, September 1992.

[14] Thomas R. Kramer. An express schema for machining plugged into alps4. Technical report, The National Institute of Standards and Technology, Gaithersburg, MD 20899, January 1991.

[15] Thomas R. Kramer. A library of material removal shape element volumes (MRSEVs). Technical Report NISTIR 4809, The National Institute of Standards and Technology, Gaithersburg, MD 20899, March 1992.

[16] Lycourgos K. Kyprianou. *Shape Classification in Computer Aided Design.* PhD thesis, Christ College, University of Cambridge, Cambridge, United Kingdom, 1980.

[17] Timo Laakko and Martti Mäntylä. Feature modelling by incremental feature recognition. *Computer Aided Design*, 25(8):479–492, August 1993.

[18] M. Marefat and R. L. Kashyap. Geometric reasoning for recognition of three-dimensional object features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):949–965, October 1990.

[19] J. Miguel Pinilla, Susan Finger, and Friedrich B. Prinz. Shape feature description using an augmented topology graph grammar. In *Proceedings NSF Engineering Design Research Conference*, pages 285–300. National Science Foundation, June 1989.

[20] S. Prabhakar and M. R. Henderson. Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models. *Computer Aided Design*, 24(7):381–393, July 1992.

[21] J. A. Reggia, D. S. Nau, and P. Y. Wang. A formal model of diagnostic inference. II. algorithmic solution and applications. *Information Sciences*, 37:257–285, 1985.

[22] William C. Regli and Dana S. Nau. Building a general approach to feature recognition of material removal shape element volumes (MRSEVs). In Jaroslaw Rossignac and Joshua Turner, editors, *Second Symposium on Solid Modeling Foundations and CAD/CAM Applications*, New York, NY 10036, USA, May 19-21, Montreal, Canada 1993. ACM SIGGRAPH, ACM Press.

[23] William C. Regli and Dana S. Nau. Recognition of volumetric features from CAD models: Problem formalization and algorithms. Technical Report TR 93-41, The University of Maryland, Institute for Systems Research, College Park, MD 20742, USA, April 1993. Submitted for publication.

[24] Scott A. Safier and Susan Finger. Parsing features in solid geometric models. In *European Conference on Artificial Intelligence*, 1990.

[25] Hiroshi Sakurai and Chia-Wei Chin. Defining and recognizing cavity and protrusion by volumes. In Fatih Kinoglu, editor, *ASME Computers in Engineering Conference*, pages 59–65, August 1993.

[26] Hiroshi Sakurai and David C. Gossard. Recognizing shape features in solid models. *IEEE Computer Graphics & Applications*, September 1990.

[27] Jami Shah, Martti Mäntylä, and Dana Nau, editors. *Advances in Feature Based Manufacturing.* Elsevier/North Holland, 1993. To appear.

[28] Jami J. Shah. Assessment of features technology. *Computer Aided Design*, 23(5):331–343, June 1991.

[29] P. Spiby. Exchange of product model data – part 11: The express language. Technical Report ISO TC184/SC4 Document N14, International Organization for Standardization, 1991. April.

[30] J. H. Vandenbrande and A. A. G. Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1269, December 1993.

[31] Jan H. Vandenbrande and Aristides A. Requicha. Geometric computations for the recognition of spatially interacting manufacturing features. In Jami Shah, Martti Mäntylä, and Dana Nau, editors, *Advances in Feature Based Manufacturing.* Elsevier/North Holland, 1993. To appear.

[32] Tony C. Woo. Feature extraction by volume decomposition. In *Conference on CAD/CAM Technology in Mechanical Engineering*, pages 76–94, March 1982.