Building MRSEV Models for CAM Applications

Satyandra K. Gupta* Thomas R. Kramer† Dana S. Nau‡#

William C. Regli†‡ Guangming Zhang*

* Department of Mechanical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742

‡ Department of Computer Science and
Institute for Systems Research
University of Maryland
College Park, MD 20742

Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

† Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Accepted for publication in *Advances in Engineering Software*.
Also available as **CS-TR-3331, UMIACS-94-97, ISR-TR93-84**.

Abstract

In integrating CAD and CAM applications, one major problem is how to interpret CAD information in a manner that makes sense for CAM. Our goal is to develop a general approach that can be used with a variety of CAD and CAM applications for the manufacture of machined parts.

In particular, we present a methodology for taking a CAD model, extracting alternative interpretations of the model as collections of MRSEVs (Material Removal Shape Element Volumes, a STEP-based library of machining features), and evaluating these interpretations to determine which one is optimal. The evaluation criteria may be defined by the user, in order to select the best interpretation for the particular application at hand.

Keywords: Design Critiquing, CAD/CAM Integration, Feature Recognition.

1 Introduction

Although various CAD and CAM applications may have compatible goals and functionality, the specific details are often different enough that it can be difficult to integrate them. One major problem is how to take information from CAD models and interpret it in a manner that makes sense for CAM. We are developing an approach to address this problem in the manufacture of machined parts. Our goal is to develop a general approach that can be used with a variety of CAD and CAM applications.

In this paper, we present a methodology for taking a CAD model, and translating it into a set of features that make sense for machining applications such as process planning, NC part programming, fixture design and selection, and manufacturability evaluation. Our approach involves extracting alternative interpretations of the CAD model as collections of volumetric features that correspond to machining operations, and evaluating these interpretations to determine which one is optimal for the particular application at hand.

Although several approaches have previously been developed for generating interpretations of parts as collections of features, 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 provided in many 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) [1]. 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 community.

2. Although many approaches have been developed for recognizing features in solid models of mechanical parts, the absence of a clear mathematical specification for the problem has made it unclear what specific classes of parts and feature interactions can be handled by various existing approaches. In particular, it has proven difficult to capture the changes that occur to feature topology and geometry when they intersect with each other in arbitrary ways.

   To address these issues, we present a formalization of the problem of 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 a set of MRSEV instances. We outline the algorithms for constructing instances of a variety of hole and pocket MRSEVs directly from geometric information in a CAD model, along with verifying accessibility constraints for those features. Our approach is guaranteed to return the complete set of MRSEV instances occurring in the alternative interpretations of the design as different collections of machinable features, even if the features intersect with each other in complex ways.

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 CAD/CAM systems lack a systematic methodology for generating and evaluating alternative interpretations of the part. 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 for its suitability to a specific CAM application.

   To address this problem, we have developed a systematic methodology capable of generating and evaluating the alternatives, in order to determine which one is optimal. The evaluation criteria may be defined by the user, in order to select the best interpretation for the particular application at hand.

Our approach works as follows:

Step 1. Recognize all MRSEVs appearing in any of the *MRSEV models* for the part. A MRSEV model is an irredundant set of MRSEVs that describe the part. We consider MRSEVs that correspond to the maximal realistic machinable volume made by a single machining operation in a single machining setup and contribute to the surface of the machined part. In most of the cases, this set of all possible MRSEVs will include redundant members.

Step 2. Generate and evaluate the alternative MRSEV models for the part, For evaluating these models, the evaluation criteria depend on the specific CAM application and its objective. For example, if the given CAM application is process planning, we may be interested in optimizing the production cost, time or profit rate. Moreover, in some CAM applications the MRSEV model should also satisfy certain additional constraints (for example, in case of process planning, the MRSEV model should result in a process plan that can achieve the desired tolerances and surface finishes).

Step 3. Return the best model, according to the evaluations done in Step 2.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents our definitions. Section 4 describes our approach for recognizing MRSEVs. Section 5 presents our algorithm for generating alternative MRSEV models for a part. Section 6 presents our algorithm for evaluating a MRSEV model. Section 7 describes our current implementation. Section 8 describes proposed future extensions of our research. Section 9 contains 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 [2, 3, 4, 5]. 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. The recent trend seems to be toward defining sets of features with specific application domains in mind (such as machining, assembly, inspection, etc.).

2.1 Recognizing Features

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 [6, 7] based on graph algorithms address a domain of polyhedral parts and features. The graph-grammar methods of [8, 9] and some approaches based on subgraph-matching may be prone to combinatorial difficulties for larger problems [10]. The recent work in [11] describes promising techniques that combat 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 [12, 13] have had success extending the capabilities of graph-based algorithms to more general $2\frac{1}{2}$-dimensional parts. Kramer [14] has presented a method for extracting non-intersecting features for a class of $2\frac{1}{2}$-dimensional parts with planar or cylindrical surfaces.

Sakurai [15] employs graph-based feature recognition techniques in combination with support for user-defined feature types. In more recent work, Sakurai and Chin [16] propose an algorithm for recognizing very general protrusions and cavities through "spatial decomposition and composition." The work of Henderson [17] was seminal in employing expert systems on the feature recognition problem.

In one of the early efforts on feature extraction, Woo [18] 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 being confined to polyhedral models and the existence of certain pathological cases in which the procedure would not converge. The non-convergence of Woo's approach has been solved in recent work by Kim [19, 20]. Kim's approach uses convex volume decompositions to produce alternating sums of volumes and techniques for partitioning the solid to avoid non-convergence. Kim further improved the approach by performing additional mapping of the volumes found to feature templates.

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 Dong [21] included the formalization of a feature description language and employed frame-based reasoning algorithms to extract machining features for computer aided process planning. An approach handling feature interactions and intersections was done by Marefat [22]. The work built on the representation scheme of [7] and used a novel combination of expert system and hypothesis testing techniques to extract surface features from polyhedral objects.

Perhaps the most comprehensive approach to date for recognizing features and handling their interactions has been that of Vandenbrande [23]. Their method is capable of finding some alternative feature interpretations and is described in the next section.

Other recent work includes feature recognition from 2D engineering drawings [24], via neural network techniques [25], for sheet-metal components [26], and feature modeling by incremental recognition [27].

2.2 Generating Alternative Feature Models

The AMPS process planning system [28] includes a "feature refinement" step, in which heuristic techniques are used to combine a set of features into a more complex feature, or split a feature into two or more features. Since the techniques are heuristic in nature, it is not entirely clear when alternative interpretations will be produced.

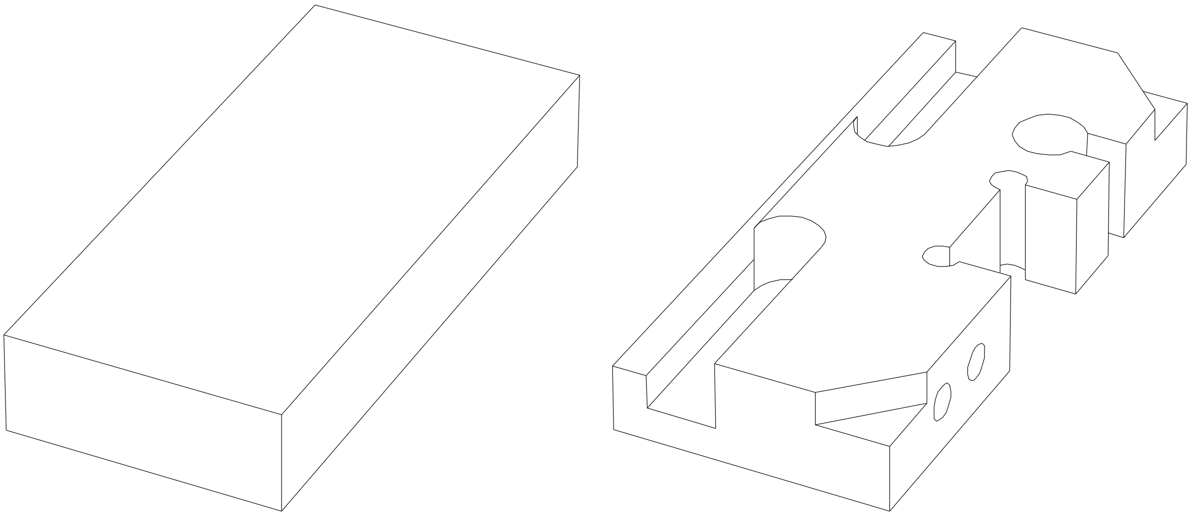
Vandenbrande [23] provides a framework for recognizing a significant class of realistic machining features of interest in process planning using artificial intelligence techniques in combination with queries to a solid modeler. He presents a set of feature classes and recognition "hints" for each class. Hints are extracted from the solid model and classified as to their potential for building feature instances. Like Dong [21], 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 and represent its interaction with other features. While the approach has many advantages, certain types of features will not be recognized if hints are removed or classified as unpromising (and thus are discarded). Further, the number of alternative feature decompositions produced is not controled.

The first systematic work in the direction of generation of alternative interpretations was done by Karinthi and Nau [29]. They described an approach for producing alternative interpretations of the same object as different collections of volumetric features as the result of algebraic operations on the features, and a system for generating alternative interpretations by performing these algebraic operations. However, this system cannot be used directly for CAM applications as there was no direct relation between these features and machining operations, hence some of the interpretations generated by this approach were not feasible from the machining point of view. Further, the algebraic operators were not sufficient to generate all interpretations of interest for machining purposes.

2.3 Evaluating Feature Models

Depending upon the specific CAM application, different evaluation functions have been developed. Extensive research has been done on different aspects of evaluation of operation plans. Mechanistic models have been developed to provide quantitative mappings from machining parameters to various performance measures, such as surface finish and dimensional accuracy [30, 28, 31, 32]. Research on machining economics has produced quantitative models for evaluating time and costs related to machining operations [33, 34].

Researchers have developed several different approaches to evaluate manufacturability [35, 36, 37, 38, 39]. Some of these have been developed for specific application domains, while others have been developed for general domains. Most of these approaches are rule-based: design characteristics which improve or degrade the manufacturability are represented as rules, which are applied to a given design in order to estimate its manufacturability.



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

Figure 1: An example part and stock.

3 Definitions and Notation

3.1 Basic Concepts

A *solid* is a manifold r-set [40] 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.

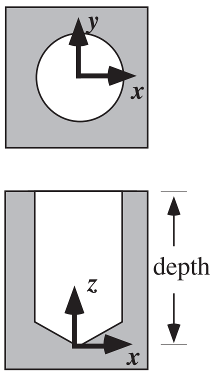
A *machined part* (or just a *part*) is the finished component to be produced as a result of a finite set of machining operations on a piece of *stock*, i.e., the raw material from which the part is to be machined. We will represent both the part and the stock as geometric solids. We use term *workpiece* 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 $\Delta = S -^* P$.

Figure 1 shows an example part and stock. Throughout this paper, we will be using this example to illustrate various steps in our approach.

3.2 Material Removal Shape Element Volumes (MRSEVs)

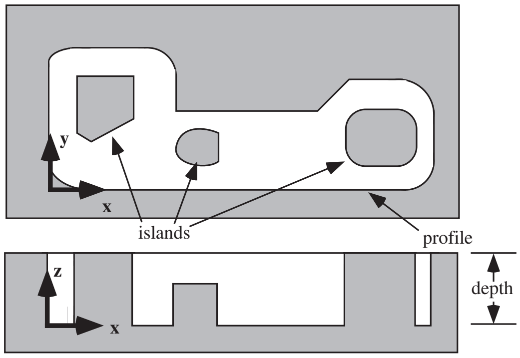
3.2.1 PDES/STEP

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



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

(a)



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

(b)

Figure 2: Subclasses of MRSEV Holes and MRSEV Pockets.

*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 [41, 42] 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 [43, 44, 45]. The EXPRESS model defines the data entities that describe the class of objects in the domain.

3.2.2 The MRSEV Hierarchy

Kramer [1, 46] 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. MRSEVs can be defined using the EXPRESS modeling language and STEP form features. Kramer 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.

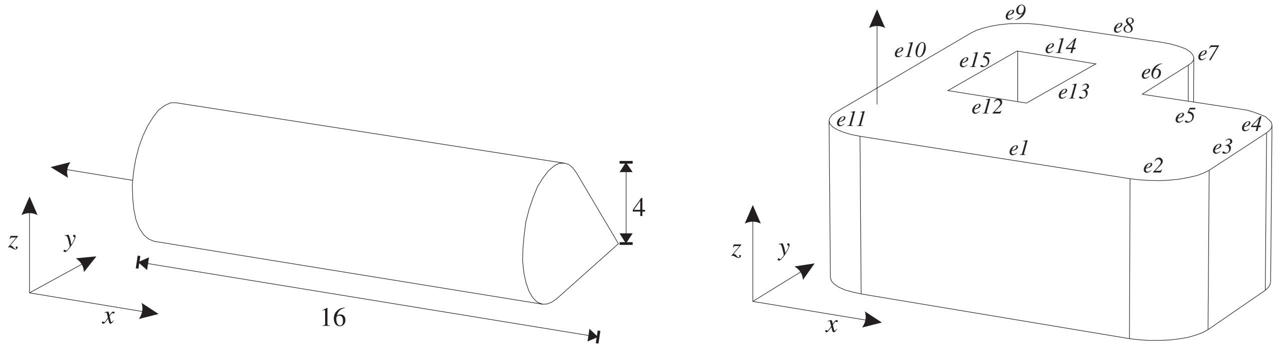
MRSEVs features are volumetric, some of the benefits of which have been explained in [47]. The MRSEV hierarchy provides a framework for describing a large class of volumetric entities of interest to machining. Each entity type has a number of required and optional attributes. MRSEV instances have been used for applications such as process planning and NC-program generation [48]. Kramer's main MRSEV types include linear swept features, edge-cut features, ramps, and rotational pockets.

For the purpose of this paper we confine our domain to a subclass of the linear swept features, i.e., conical bottomed holes with fixed tip angle of 120 degrees, general profile pockets, and pockets with islands. 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. In the case of a pocket with islands, an island is considered to be a subfeature defined by its own closed profile. Figs. 2(a) and 2(b) present our illustrations of hole and pocket MRSEVs.

3.2.3 MRSEV Instances

The MRSEVs are parameterized solids. A specific instance of one of these MRSEVs can be instantiated by assigning specific choice of attribute values. For example, suppose we choose the following attribute values:

$$\begin{array}{lcl} \textbf{location} & = & (16, 10, 4); \\ \textbf{orientation} & = & (-1, 0, 0); \\ \textbf{depth} & = & 16; \\ \textbf{radius} & = & 4. \end{array}$$



(a) A MRSEV hole.

(b) A MRSEV pocket.

Figure 3: MRSEV instances.

This would define the conical-bottomed hole illustrated in Figure 3(a). Similarly, the following values would define a MRSEV pocket with a single island as pictured in Figure 3(b):

$$
\begin{aligned}
\textbf{location} &= (0,0,1);\\
\textbf{orientation} &= (0,1,0);\\
\textbf{depth} &= 2;\\
\textbf{profile} &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\};\\
\textbf{islands} &= \{I_1\};\\
\textbf{profile } I_1 &= \{e_{12}, e_{13}, e_{14}, e_{15}\};\\
\textbf{height } I_1 &= 2.
\end{aligned}
$$

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 $b(T)$ is naturally partitioned into three pieces, as shown in Figure 4(a):

- the *separation surface* $s(T)$, i.e., the portion of $b(T)$ that connects to the rest of the machine tool;
- the *cutting surface* $c(T)$, i.e., the portion of $b(T)$ that is capable of cutting metal;
- the *non-cutting surface* $n(T)$, i.e., the portion of $b(T)$ that is not capable of cutting metal.

For the purpose of locating the tool, we will choose a particular point p_{td} of T as a *datum point*. Usually p_{td} will be the tip of the cutting-tool volume, but not always.

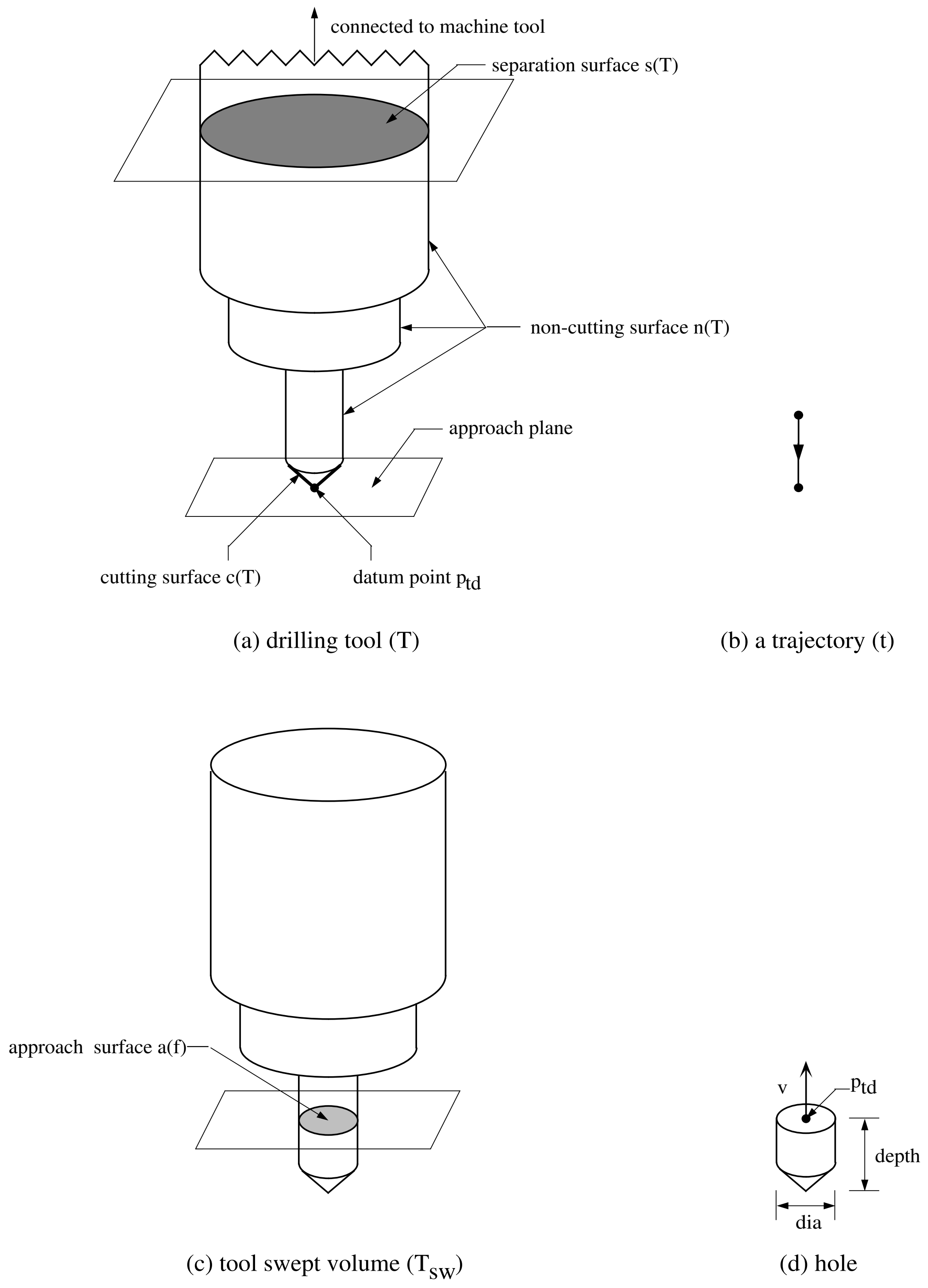


Figure 4: A cutting tool, and the resulting removal volume.

To perform the machining operation, one sweeps the tool volume T along some trajectory t, as shown in Figure 4(b). Given a tool T and a workpiece W, the trajectory t is *feasible* for T and W only if sweeping T along t does not cause interference problems between the non-cutting surface $n(T)$ and the workpiece. If t is feasible, then the volume created by sweeping T is

$$T_{sw} = \{(p - p_{td}) + q : p \in T \text{ and } q \in t\},$$

as shown in Figure 4(c). Now, let π be the plane perpendicular to t at the point p_{td}, as shown in Figure 4(a). Then the solid consisting of all points in T_{sw} that are on or below π represents the material which can be removed by the machining operations. The solid shown in Figure 4 (d) represents the volume which can be removed by a drilling operation. We can use MRSEVs to represent volumes which can be removed during machining. In particular, a MRSEV hole represents the volume which can be removed by a drilling operation, and a MRSEV pocket represents the 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 slots, steps, profiles, slabs, etc.

3.4 Effective Volume of a MRSEV

The volume removed by a MRSEV m from a given workpiece W is not necessarily m's volume. Instead, it is m's *effective volume with respect to* W, which is defined as $\text{eff}(m, W) = W \cap^* m$. Figure 5 shows a pocket MRSEV and its effective volume with respect to the workpiece.

3.5 Truncation of a MRSEV

Truncation of a MRSEV m with respect to a solid W returns the smallest MRSEV n of m's type and orientation such that n can remove the volume removed by m from W, i.e., $\text{eff}(n, W) = \text{eff}(m, W)$. An example of MRSEV truncation is shown in Fig 6.

3.6 MRSEV Models.

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

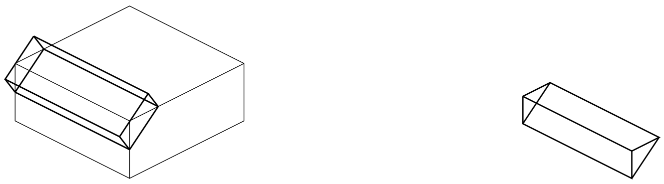
1. If we subtract the MRSEVs in M from S, we get P; i.e., $S -^* \cup^*_{m \in M} m = P$.

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 MRSEV model is an interpretation of the delta volume as a set of machining features. For example, the set $\{h1, h2, s1, s2\}$ shown in Figure 7 is a MRSEV model.

4 Recognizing MRSEVs

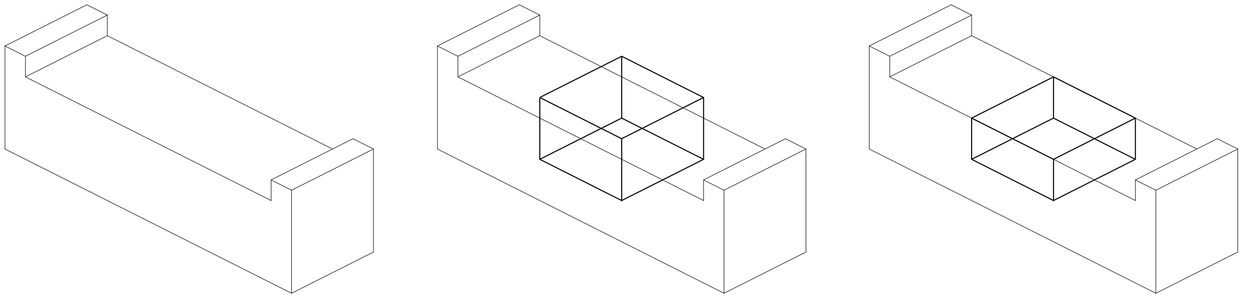
Given solids representing the part P and the stock S, we are interested in finding the set of all MRSEVs that can be used in a MRSEV model of the part. In this section we present a methodology for recognizing instances of hole and pocket MRSEVs.



(a): a MRSEV, and its effect on the workpiece

(c): effective volume with respect to the workpiece

Figure 5: A pocket MRSEV and its effective volume.

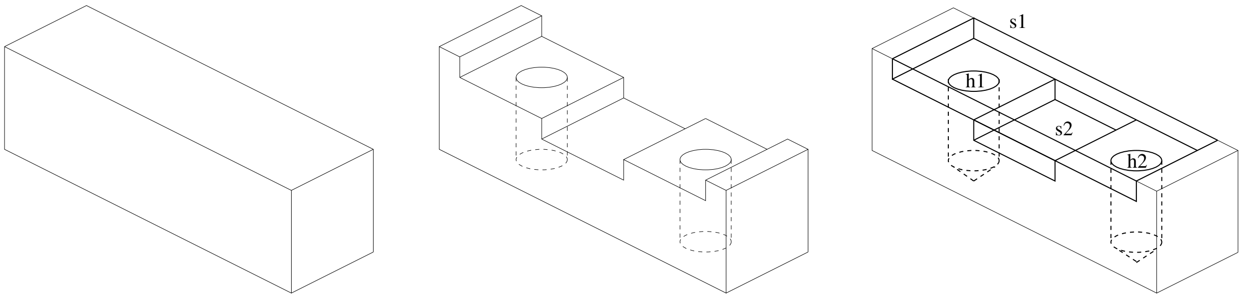


(a):workpiece

(b): before truncation

(c): after truncation

Figure 6: An example of MRSEV truncation.

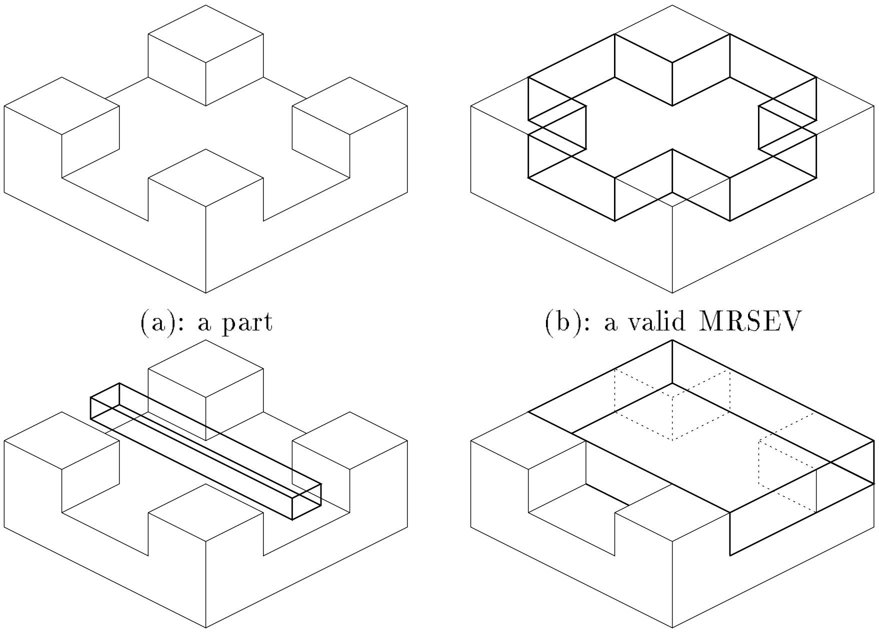


(a): stock (before machining)

(b): part (after machining)

(c) MRSEV model

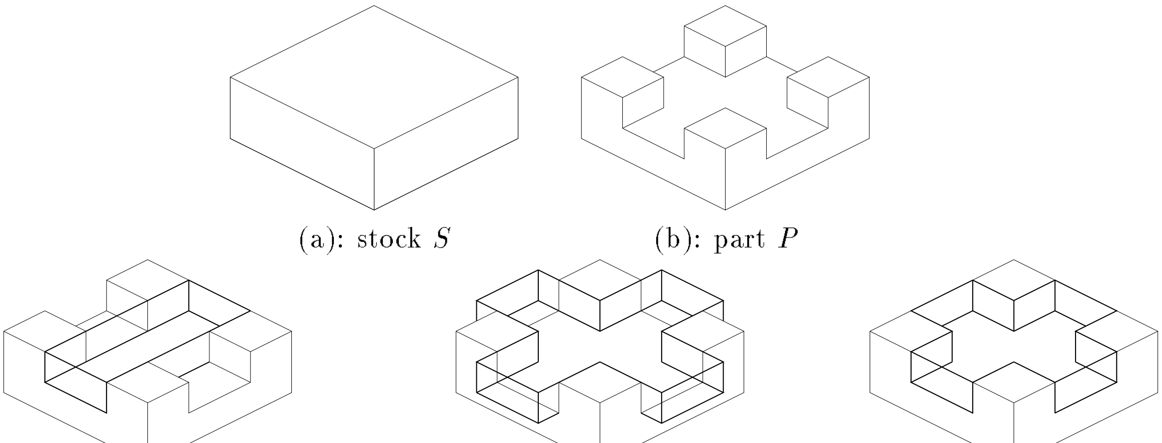
Figure 7: An example of a MRSEV model.



(a): a part (b): a valid MRSEV

(c): an invalid MRSEV (d): an invalid MRSEV

Figure 8: Examples of valid and invalid MRSEVs.



(a): stock S (b): part P

(c): non-primary (d): non-primary (e): primary

Figure 9: Example of primary and non-primary MRSEVs.

The input is a pair of solids representing the initial stock S, and the final part P. Given a realistic part, there will be a large (theoretically speaking, infinite) number of MRSEVs that may be used to describe the part, each corresponding to machining operations that might be used to create it. In general, a feature recognition algorithm is *complete* if, for all P and S, if it returns the set of all valid features that can be used to produce P from S. We will only be interested in recognizing those MRSEVs which will be useful in CAM applications. Therefore, we define the following restrictions on the 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., $m \cap^* P = \emptyset$).

Figure 8 shows examples of valid and invalid MRSEVs.

Primary MRSEVs. A *primary* MRSEV for a part P and stock S is any valid MRSEV m, such as those illustrated in Figure 9, 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 $\text{eff}(n, S) \subset \text{eff}(m, S)$).

An arbitrary part and stock may still present a problem because they may give rise to an infinite number of possible primary MRSEVs. For realistic parts, most of these possibilities will not correspond to reasonable machining operations and removal volumes. We will consider the class of primary MRSEVs with the following characteristics:

1. For any primary hole MRSEV h, the delta volume contains either a subface of h's cylindrical side surface or h's entire ending surface;
2. For any primary pocket MRSEV p, either a subface of p's bottom face is present in the delta volume, or p is a through pocket with subfaces of two or more non-parallel planar side faces or one non-planar side face present in the delta volume.

The output of the MRSEV recognizer is a primary *MRSEV set*, $\mathcal{M}$: a finite set whose elements are all primary MRSEVs satisfying the characteristics enumerated above. In the context of this paper, a MRSEV recognition algorithm is complete if it returns the set of all primary MRSEV instances with these properties that appear in any of the MRSEV models of P and S (if no such MRSEV models exist, the algorithm reports so).

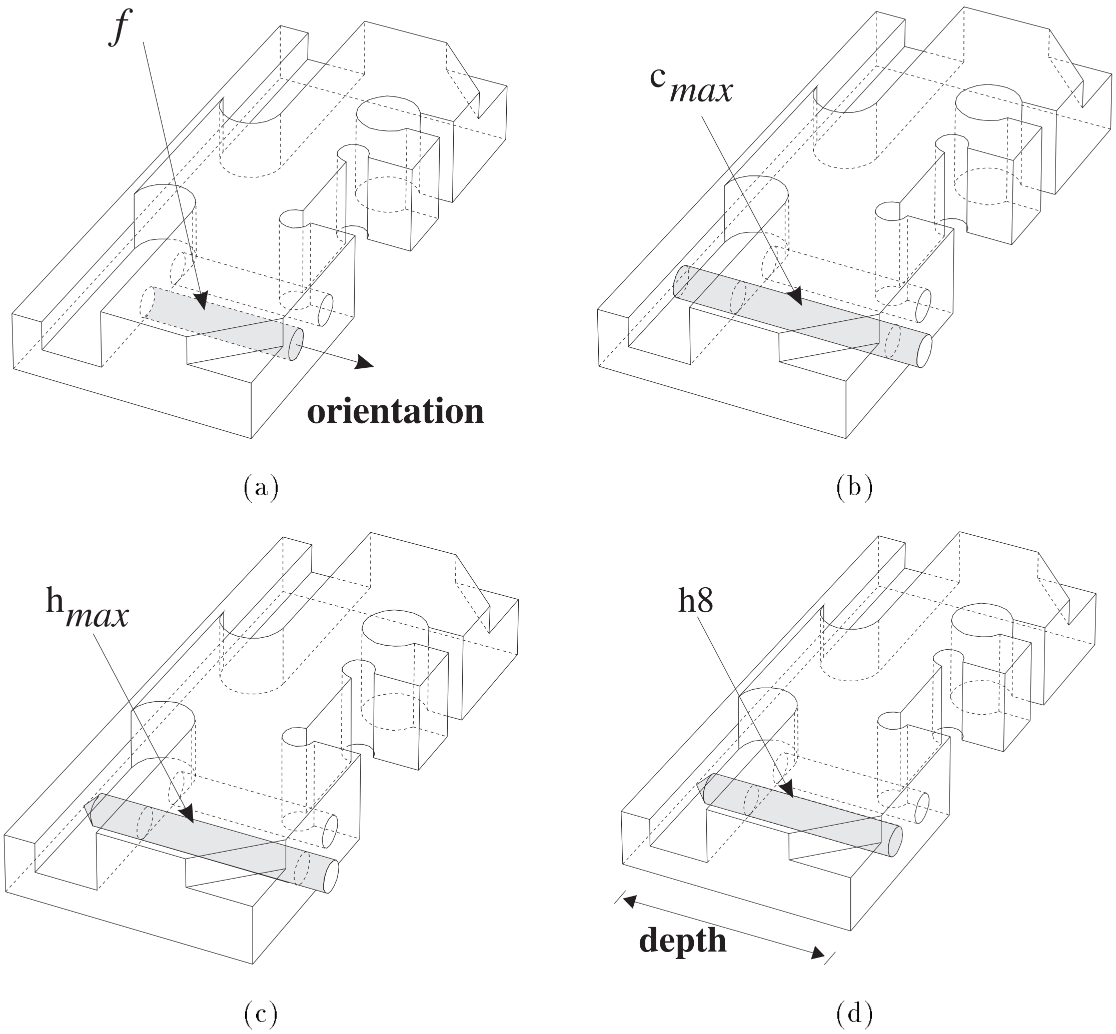


Figure 10: Construction of a MRSEV hole.

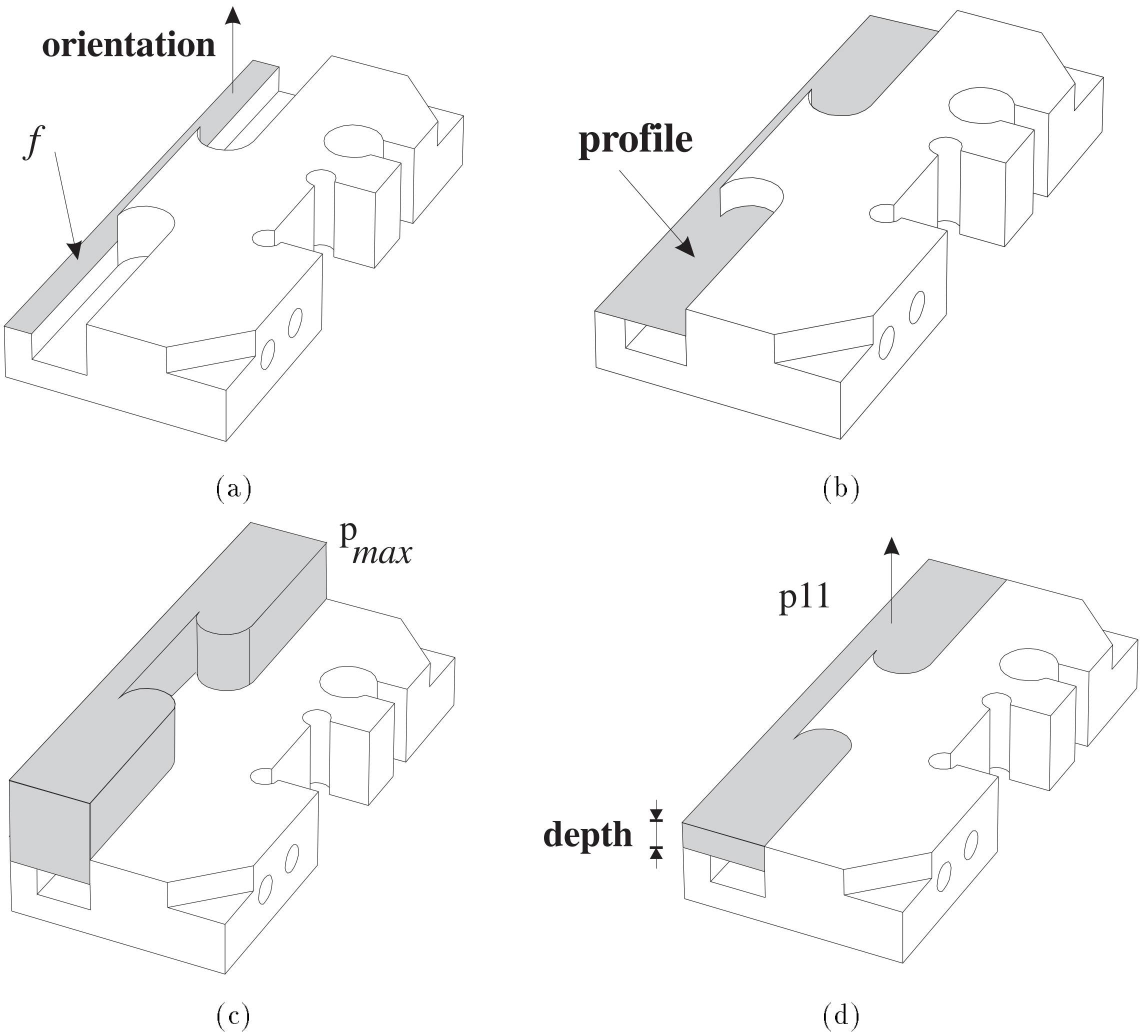


Figure 11: Construction of a MRSEV pocket.

4.1 MRSEV Recognition

Given the specifications for the MRSEV recognition problem, we can outline the algorithms for solving it. We start with a solid model of the part P and construct all instances of primary MRSEVs that can be built from the geometric and topological information contained in the boundary-representation [49] of P. Proceeding from the observation that every valid primary MRSEV instance m must contribute to some face of the delta volume, the set of primary MRSEV instances can be found by traversing the faces of the delta volume and instantiating those primary MRSEVs capable of covering all or a portion of each face. A high-level description of the MRSEV recognition algorithm can be given as follows:

RECOGNIZE_MRSEVS:
INPUT: solid models of a part P and stock S
OUTPUT: a primary MRSEV set, $\mathcal{M}$.

1. For each face f of $S -^* P$ do:
2. If f is a concave cylindrical face, f might be subface of the side of a MRSEV hole or a subface of a round side face of a through pocket. Construct the possible primary instances of MRSEVs that might have created f as described below in RECOGNIZE_HOLES and RECOGNIZE_POCKETS.

   If f is a convex cylindrical face, f might be a subface of a round side face of a through pocket. Construct the instances of primary MRSEVs capable of creating f (RECOGNIZE_POCKETS).
3. If f is a planar face, f might be a subface of the bottom surface of a non-through pocket or a subface of a side surface of a through pocket (RECOGNIZE_POCKETS).
4. If f is a concave conical face, f may be the end surface of a hole (RECOGNIZE_HOLES).
5. return, $\mathcal{M}$, the set of features built.

Depending on the type of surface, we calculate a parameterization for each possible primary MRSEV that might have created it. In 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. As space does not permit full elaboration on the geometric details of constructing primary MRSEVs, we present pseudo-code outlines of the structure of these algorithms, as their implementation will depend greatly on the functionality of the modeling system being employed [50, 51].

RECOGNIZE_HOLES: Recognizing hole MRSEVs is straightforward: an instance of a hole can be found from its end surface face or a portion of its cylindrical side face (as shown in Figure 10(a)). For non-through hole features, only one feature instance exists. In the case of a cylindrical face produced by hole that extends through the part, there are two possible orientations of a primary feature instance: one in each direction along the axis of the cylindrical surface. For a given face f, the steps in this process are:

1. Confirm that face f is potential subface of a MRSEV hole, i.e. f is a cylindrical or conical face. Values for radius and orientation parameters can be found from f, as shown in Figure 10(a) where f is a cylindrical face that is part of the side of a MRSEV hole.

2. Find the maximal[1] non-intrusive cylinder c_{max}, as shown in Figure 10(b), and determine f's accessibility i.e. c_{max} must extend beyond the stock in at least one direction along the orientation.

3. If f can be made as a surface of a hole, determine a location for a maximal MRSEV hole, h_{max} as shown in Figure 10(c). In the case of a through hole, locations outside the stock can be chosen for each of the two maximal MRSEV holes. In the case where f has not been made by a through hole, there are two possibilities for locating the MRSEV hole: f is a conical surface and is itself the bottom face; or the tip of the conical end face is located on the planar side face of c_{max}. Note there exist situations where this yields an approximation of a primary feature. For purposes of machining, this approximation produces satisfactory results.

4. Truncate h_{max} to get the instance of the primary MRSEV hole, as shown in Figure 10(d) for h_8. In the event that f is accessible bi-directionally, there will be two instances of primary MRSEV holes, as shown in Figure 12 for holes h_5 and h_6.

RECOGNIZE_POCKETS: Construction of pocket features starts at a face of the delta volume, f. For each face f in the part P created by an instance of a pocket MRSEV, there are two possibilities:

1. A face f could be a subface of the planar bottom surface of a pocket MRSEV, as shown in Figure 11(a).

2. A face f could be a subface of a side face of a pocket MRSEV which extends through the part, possibly a corner radius or a pocket wall. This type of feature is often called a through pocket, an example of which is feature p_7 shown in Figure 13.

In the first case, an orientation for the pocket MRSEV is determined from the surface normal of f, as shown in Figs. 11(a). In the second case where the feature is a through pocket, there are two possible MRSEV pocket instances having opposite orientations. These orientations can be determined from either the axis of the cylindrical surface or the cross product of the normal vectors of f and another planar surface f' elsewhere in the delta volume.

For this class primary MRSEV features, the pocket profile can be computed from the projection of the part faces that lie above (with respect to the orientation) the plane containing the bottom surface of the pocket, as illustrated in Figure 11(b), and an arbitrary location for the pocket based on the profile chosen. In the second case, where the pocket extends through the part and there is no bottom surface present in the delta volume, an arbitrary location can be chosen for the projection plane and the all of the part faces are mapped onto it. In this way,

[1] Mathematically, "maximal" would imply a cylinder of infinite length, but in practice it is sufficient to extend the cylinder to any point beyond the stock.

we ensure that the MRSEV pocket is accessible in the direction of its orientation and calculate the maximal pocket profile capable of creating these surfaces.

Given the profile, an instance of a maximal MRSEV pocket p_{max} can be created, as shown in Figure 11(c). In the case of a through pocket, two maximal MRSEV pocket instances are created. We truncate p_{max} to obtain the primary MRSEV pocket, as shown in Figure 11(d), with a depth sufficient to extend the feature instance outside the stock. Features p_7 and p_8 in Figure 13 show examples of through pocket MRSEVs.

The MRSEV holes found for the part in Figure 1 are shown in Figure 12.

Example. For the part in Figure 1, Figs 12 and 13 shows the various MRSEVs identified by our algorithm. In this case, MRSEV set is

$$\mathcal{M} = \{p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, h1, h2, h3, h4, h5, h6, h7, h8\}.$$

5 Building MRSEV Models from a MRSEV Set

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 3.6, 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 MRSEV model, each one corresponding to a potential way of making the part. In this section, we present an algorithm for selecting the best MRSEV model for a given CAM application.

5.1 Algorithm for Generating Best MRSEV Model

Each MRSEV model is basically a set cover[2] of the delta volume $\Delta = S -^* P$. We shall employ a set-covering algorithm to generate MRSEV models and use pruning heuristics to discard the unpromising MRSEV models. The algorithms for this are:

FIND_BEST_MODEL	Assigns initial values to some variables, and calls FIND_COVERS to generate-and-test various MRSEV models.
FIND_COVERS	A backtracking algorithm that looks for sets of effective volumes that form irredundant set-covers for the delta volume. Each such set cover corresponds to one or more MRSEV models.
GENERATE_MODELS	For each of the irredundant covers found by FIND_COVERS, this algorithm finds one or more MRSEV Models M such that

[2]The *set covering problem* is a well known combinatorial problem [52]. There are several different variations of this problem; the one that is most appropriate for our purposes is the following [53, 54]. Let S be a set, and $\mathcal{S} = \{S_1, \ldots, S_n\}$ be a family of sets. Find all possible irredundant subsets $\mathcal{T} \subseteq \mathcal{S}$ such that $S \subseteq \bigcup(\mathcal{T})$. By irredundant, we mean that there is no proper subset $\mathcal{U} \subset \mathcal{T}$ such that $S \subseteq \bigcup(\mathcal{U})$.

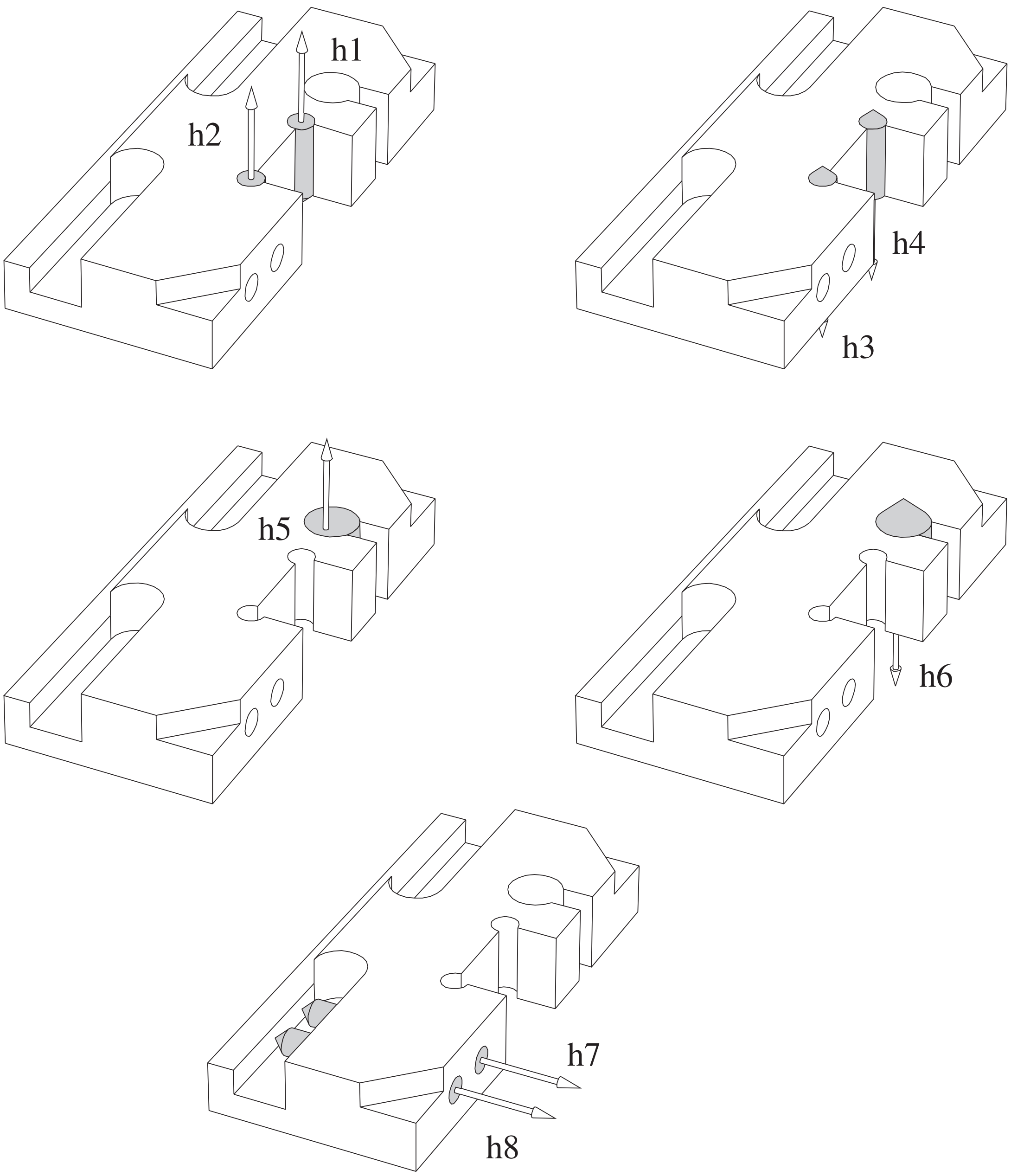


Figure 12: MRSEV holes identified for the part shown in Figure 1.

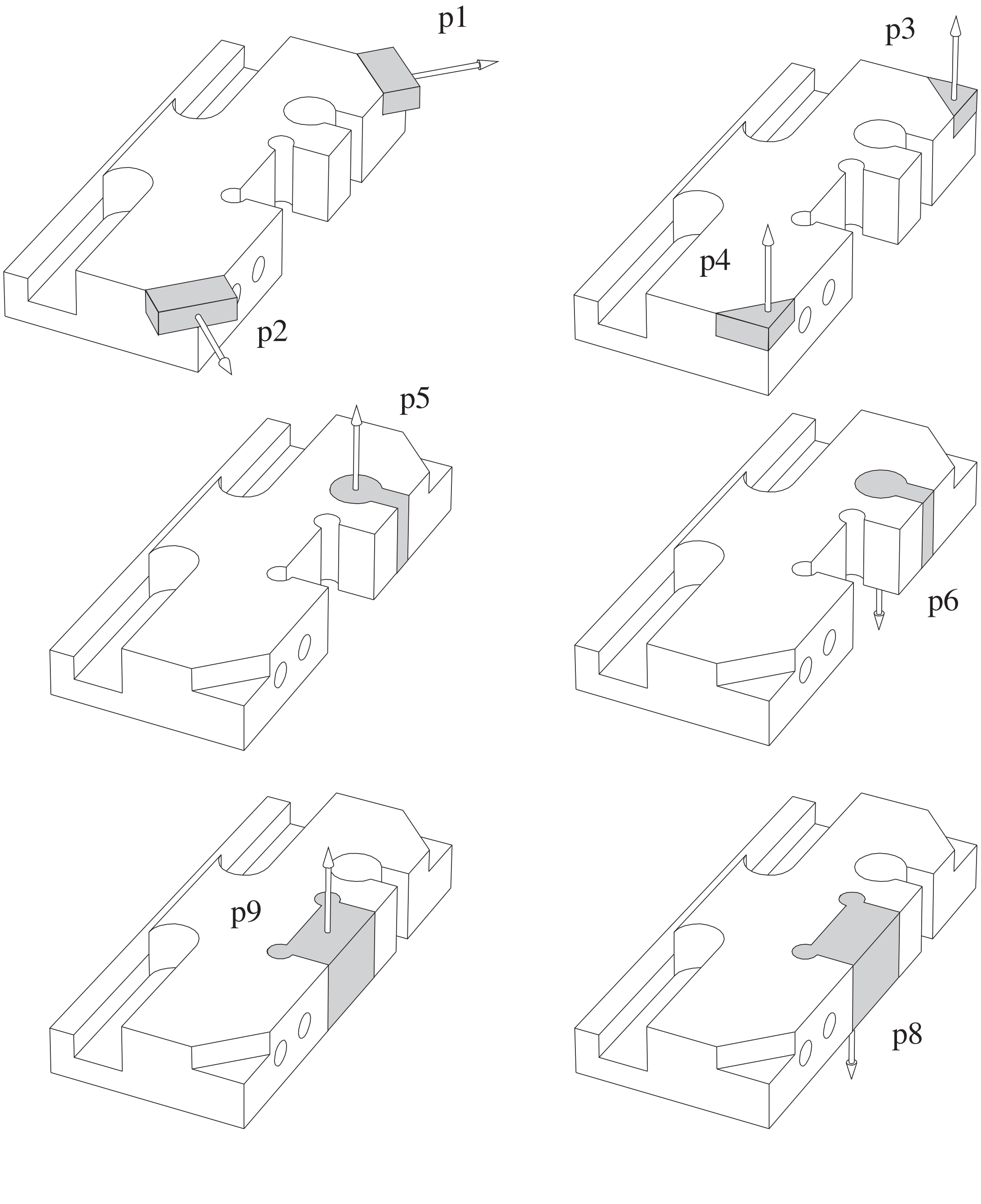


Figure 13: MRSEV pockets identified for the part shown in Figure 1.

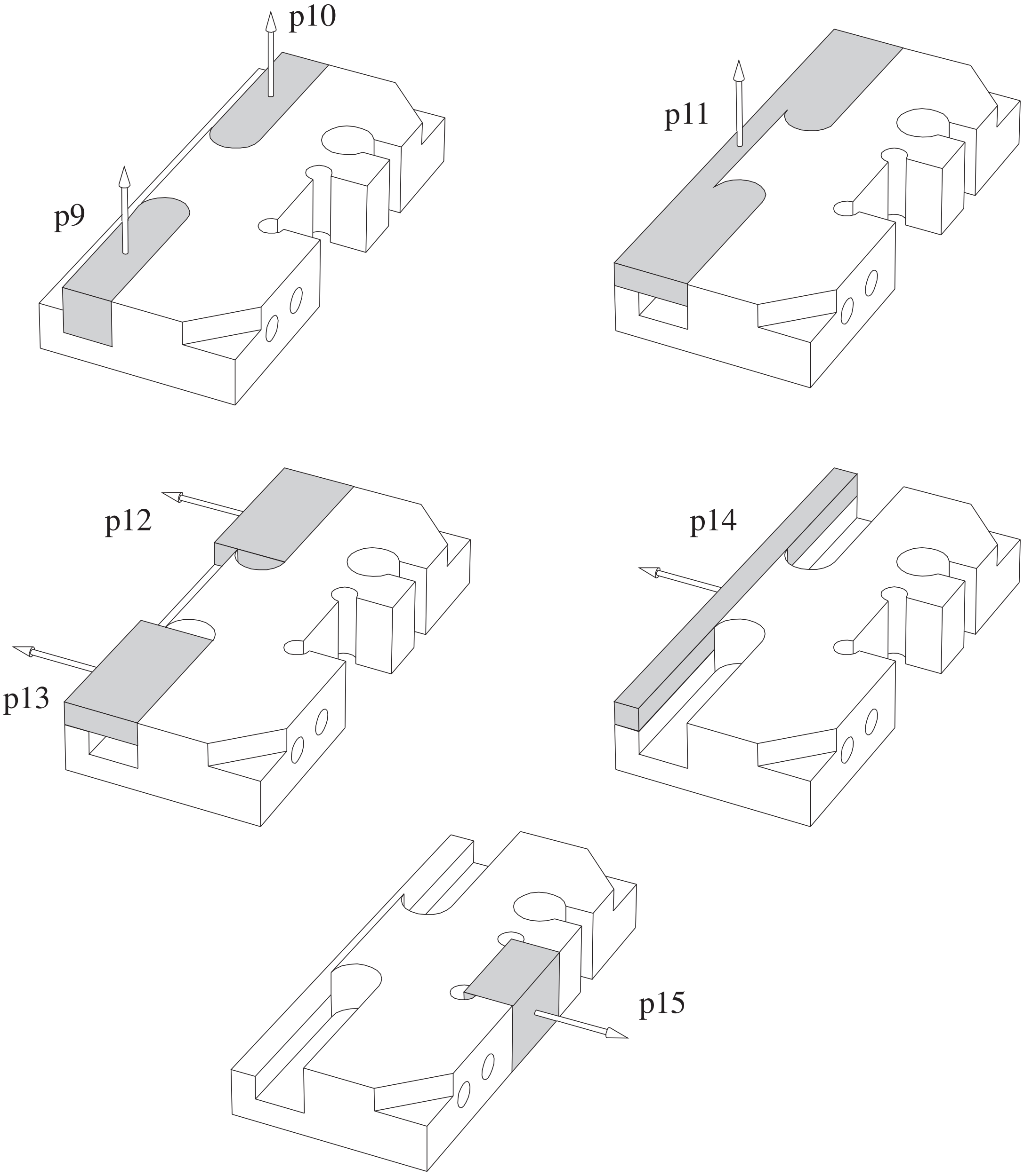


Figure 13 continued.

the effective volumes of the MRSEVs in M are identical to the volumes in the irredundant cover.

Whenever GENERATE_MODELS finds a MRSEV model, it evaluates it for the given CAM application and compares it with the current best model.

EVALUATE_MODEL	Evaluates a MRSEV model for a specific CAM application–described in the next section.

First we present the FIND_BEST_MODEL algorithm. This algorithm computes the set $\mathcal{V}$ of effective volumes with respect to the stock S, and then splits $\mathcal{V}$ into two parts. One part, V, contains each volume that is not subsumed by the other volumes in $\mathcal{V}$. These volumes are guaranteed to be in every irredundant cover for $\mathcal{V}$. The other part, $\mathcal{V} - V$, contains each volume that is subsumed by the other volumes in $\mathcal{V}$. These volumes may appear in some irredundant covers for $\mathcal{V}$, but will not appear in all of them. To compute the irredundant covers and find the best one, FIND_BEST_MODEL invokes a subroutine called FIND_COVERS.

Algorithm 5.1 FIND_BEST_MODEL($\mathcal{M}$)
INPUT: *a primary MRSEV set,* $\mathcal{M}$.
OUTPUT: **best_MRSEV_model**, *the best MRSEV model as calculated by an evaluation function.*

$\mathcal{V} = \{\mathrm{eff}(m, S) : m \in \mathcal{M}\}$
$V = \{v : v -^* \bigcup_{q \in \mathcal{V} - \{v\}}(q) \neq \emptyset\}$
best_model $= \emptyset$
best_eval $= \infty$
`for every` $C \in$ FIND_COVERS($\mathcal{V} - V, V$)`, do`
  (**best_eval**, **best_model**) = GENERATE_MODELS($C, \emptyset$, **best_eval**, **best_model**)
`return(`**best_MRSEV_model**`)`

For the MRSEVs shown in Figs. 12 and 13, the set of effective volumes with respect to the stock is:

$$\mathcal{V} = \{v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16\},$$

where

$$\begin{array}{ll}
v1 = \mathrm{eff}(p1, S) = \mathrm{eff}(p3, S), & v2 = \mathrm{eff}(p2, S) = \mathrm{eff}(p4, S), \\
v3 = \mathrm{eff}(p5, S) = \mathrm{eff}(p6, S), & v4 = \mathrm{eff}(p7, S) = \mathrm{eff}(p8, S), \\
v5 = \mathrm{eff}(p9, S), & v6 = \mathrm{eff}(p10, S), \\
v7 = \mathrm{eff}(p11, S), & v8 = \mathrm{eff}(p12, S), \\
v9 = \mathrm{eff}(p13, S), & v10 = \mathrm{eff}(p14, S), \\
v11 = \mathrm{eff}(p15, S), & v12 = \mathrm{eff}(h1, S) = \mathrm{eff}(h3, S), \\
v13 = \mathrm{eff}(h2, S) = \mathrm{eff}(h4, S), & v14 = \mathrm{eff}(h5, S) = \mathrm{eff}(h6, S), \\
v15 = \mathrm{eff}(h7, S), & v16 = \mathrm{eff}(h8, S).
\end{array}$$

In this case, V is the set $\{v1, v2, v3, v5, v6, v15, v16\}$.

The FIND_COVERS algorithm takes two arguments, X and V. V is the partial set cover that is being built up and X is a set of volumes that can potentially be added to V to complete the set cover. FIND_COVERS calls itself recursively, removing elements from X and adding them to V. Upon entry, if X contains a complete cover, FIND_MRSEVs is called. For example, FIND_BEST_MODEL calls FIND_COVERS with $X = \{v4, v7, v8, v9, v10, v11, v12, v13, v14\}$ and $V = \{v1, v2, v3, v5, v6, v15, v16\}$.

In cases where X is nonempty, the efficiency of FIND_COVERS will depend on the order in which it chooses the volumes in X. To make the procedure more efficient, our heuristic is to choose the volume v in X that covers the maximum portion of the remaining delta volume (i.e., choose a $v \in X$ such that $v \cap^* (\Delta -^* \cup(V))$ is maximized).

Algorithm 5.2 FIND_COVERS(X, V)
INPUT: *V, a partial set cover; X a set of volumes.*
OUTPUT: *a set of irredundant covers of the delta volume.*

```
if V contains a volume v subsumed by the other volumes in V
    (i.e., ⋃(V − {v}) = ⋃V)
    return ∅ // V is redundant
if the delta volume is completely covered by V
    (i.e., Δ ⊆ ⋃V)
    return {V}// we have found an irredundant cover
if the volumes in V and X cannot cover the delta volume
    (i.e., Δ ⊈ ⋃(V ∪ X))
    return ∅ // V is not feasible
choose a volume v in X
return(FIND_COVERS(X − {v}, V ∪ {v}) ⋃ FIND_COVERS(X − {v}, V))
```

For the MRSEVs shown in Figs. 12 and 13, FIND_COVERS finds following four covers:

$$V1 = \{v1, v2, v3, v4, v5, v6, v10, v15, v16\};$$
$$V2 = \{v1, v2, v3, v4, v5, v6, v7, v15, v16\};$$
$$V3 = \{v1, v2, v3, v5, v6, v7, v11, v12, v13, v15, v16\};$$
$$V4 = \{v1, v2, v3, v5, v6, v10, v11, v12, v13, v15, v16\}.$$

Each time that FIND_COVERS finds an irredundant cover for the delta volume, the next step is to generate one or more MRSEV models from this cover. This is done by using the depth-first branch-and-bound algorithm GENERATE_MODELS described below. GENERATE_MODELS takes four arguments, V and N, **best_model** and **best_eval**. N is the partial MRSEV model that has been built up already, V is the set of volumes from which MRSEVs need to be generated in order to finish N's cover, **best_model** is the best MRSEV model that has been seen so far, and **best_eval** is its evaluation function value. GENERATE_MODELS is called recursively to remove volumes from V, and to explore alternative completions of N's cover. For each

MRSEV model that GENERATE_MODELS generates, it evaluates the MRSEV model by calling EVALUATE_MODEL described in the next section.

If good MRSEV models are generated and examined first, then we need not examine any MRSEV model that is not expected to be better than the current best. We use heuristic $h(N, V)$ to estimate the lower bound of the evaluation function value. This heuristic depends on the particular CAM application. An example of such a heuristic is described in the next section.

Algorithm 5.3 GENERATE_MODELS($V, N,$ **best_eval**, **best_model**)
INPUT: V, *a partial set cover;* N *a partial MRSEV model,* **best_eval,best_model**.
OUTPUT: **best_eval**; **best_model**

```
if h(N,V) ≥best_eval
   return (best_eval,best_model)
          // N is unpromising
          // The pruning heuristic h(N,V) estimates the lower
          // bound of best_eval resulting
          // from MRSEVs in set N. This heuristic is described
          // in the next section.
if V = ∅
   // we have found a MRSEV model
   if EVALUATE_MODEL(N) < best_eval
          // EVALUATE_MODEL returns evaluation function values
          // for a specific application domain. How this evaluation
          // is performed is described in the next section.
      best_eval = EVALUATE_MODEL(N)
      best_model = N
   return (best_eval,best_model)
else
   choose a volume v in V
   let Assc be the set of all MRSEVs in M having
      v as their effective volume (i.e., Assc = {l : eff(l,S) = v})
   for each MRSEV n ∈ Assc
      (best_eval,best_model) =GENERATE_MODELS(V − {v}, N ∪ {n},
                                                 best_eval,best_model)
   return(best_eval,best_model)
```

The efficiency (but not the correctness) of GENERATE_MODELS depends on the order in which which volumes v are chosen from V. Our heuristic is to choose the one that has the minimum number of MRSEVs associated with it, i.e., to choose $v \in V$ that minimizes the cardinality of the set $\{l : \text{eff}(l, S) = v\}$. The efficiency also depends on the order in which it examines the MRSEVs in $\mathcal{A}ssc$. Our heuristic is examine MRSEVs $n \in \mathcal{A}ssc$ in order of increasing value of the pruning heuristic $h(N \cup \{n\}, V - \{v\})$.

Example. For the MRSEVs shown in Figs. 12 and 13, the MRSEV model

$$M = \{p3, p4, p5, p9, p10, p11, p15, h1, h2, h7, h8\}$$

produces the lowest value of the evaluation function (described in next section). This model was generated from cover $V3$.

6 Evaluating MRSEV Models

Depending upon the CAM application, we are given some evaluation function. In most of the cases, we are interested in finding the MRSEV model which optimizes the value of this evaluation function. For example, if we want to use the MRSEV model for process planning, our evaluation function could be production cost, production time or a combination of these.

Besides optimizing the evaluation function value, most CAM applications will require that the MRSEV model should satisfy some additional constraints. For example, in the case of process planning, operations associated with the MRSEV 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.

Each machining operation creates a MRSEV which has certain geometric variations compared to its nominal geometry. Designers normally give tolerance specifications on the nominal geometry, to specify how large these variations are allowed to be. 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. For the sake of brevity we will not describe tolerance estimation in this paper. For details on our work on tolerance estimation, readers are referred to [55, 56].

It is also worth noticing that MRSEVs in a MRSEV model cannot necessarily be machined in any arbitrary order. Instead, accessibility [56], tolerance-datum dependencies, setup [57] and other types of interactions among them will introduce *precedence constraints* requiring that some of them be machined before or after others. Discussing various types of precedence constraints is out of the scope of this paper. Our work on identifying precedence constraints can be found in [56].

In most CAM applications, a general evaluation framework will require following three steps:

1. Perform pre-processing of the MRSEV model to identify precedence constraints on the MRSEVs in the model.
2. Verify that the MRSEV model satisfies the domain specific constraints.
3. If the MRSEV Model violates any domain specific constraint, then set the value of evaluation function to infinity. Otherwise, estimate the value of the evaluation function for the MRSEV model.

Since the main focus if this paper is to present a generalized framework for building MRSEV models, we will not discuss the specifics of first two steps of the evaluation framework. The following section describes an example evaluation function for process planning applications.

6.1 An Example Evaluation Function: Production Time

Production time is quite often used to judge the merit of a process plan. Three main components of production time are the actual machining time (when the tool engages in cutting), auxiliary time and setup time (time spent in setting up the workpiece on the machining center).

The following algorithm was used to evaluate various MRSEV models in the previous section.

Algorithm 6.1 EVALUATE_MODEL(M)
INPUT: *a MRSEV model M.*
OUTPUT: *production time associated with M.*

```
use machining heuristics to determine precedence
   among interacting features
let m_1, m_2, ..., m_n be a total ordering of M that is
   consistent with the precedence constraints
   determined in the previous step
for every i > 0
   let n_i be the truncation of m_i with respect
      to the workpiece W_i = S -* (m_1 ∪* ... ∪* m_{i-1}).
let N = {n_1, n_2, ..., n_n}
calculate PT(N) // using formula described below.
return PT(N)
```

[30, 28] describe various types of precedence constraints resulting from machining considerations.

For the truncated MRSEV model N, production time is computed by the following formula:

$$\begin{aligned}
PT(N) &= \sum_{i=1}^{m} T_{si} + \sum_{n \in N} (1+\beta) T(n); \\
T_{si} &= \text{required time for setup } i; \\
m &= \text{the minimum possible number of setups (for three-axis machining centers, } m \text{ is the} \\
&\quad \text{cardinality of the set } \{\vec{o}(n) : n \in N\}\text{, where } \vec{o}(n) \text{ is the unit orientation vector for} \\
&\quad \text{MRSEV } n); \\
T(n) &= \text{the machining time associated with MRSEV } n \text{ (ways to estimate the machining} \\
&\quad \text{time for various machining operations are described in [58, 59, 33, 60]);} \\
\beta &= \text{an estimate of the auxiliary time as a fraction of the machining time (we use } \beta = 0.2).
\end{aligned}$$

To compute $PT(N)$, we approximate the quantity $\sum_{i=1}^{m} T_{si}$ by using $m \times T_s$, where T_s is the average setup time, estimated using information from handbooks such as [59].

6.2 Pruning Heuristic $h(N, V)$ for Estimating Lower Bound on Production Time

We define the heuristic function $h(N, V)$ to give the lower bound on the required setup and machining time for any operation plan resulting from MRSEVs in N.

Each time that GENERATE_MODELS is called, V is a set of effective volumes, and N is a set of MRSEVs such that $V \cup \{\text{eff}(n, S) : n \in N\}$ is an irredundant cover for the delta volume. For all sets V and N that satisfy this property, the heuristic $h(N, V)$ is defined as:

$$
\begin{aligned}
h(N,V) &= L_s(N) \times T_s + \sum_{n \in N} (1+\beta) L_T(n); \\
T_s &= \text{average setup time;} \\
L_s(N) &= \text{lower bound on the number of setups needed to machine } N \text{ (for three-axis machining} \\
&\quad \text{centers, } L_s(N) \text{ is the cardinality of the set } \{\vec{o}(n) : n \in N\}\text{, where } \vec{o}(n) \text{ is the unit} \\
&\quad \text{orientation vector for MRSEV } n\text{);} \\
L_T(n) &= \text{lower bound on the time required to machine MRSEV } n \text{ (this is the time required} \\
&\quad \text{to machine the irredundant portion of the effective volume of } n \\
&\quad \text{(i.e., eff}(n,S) -^* \cup(V) -^* \cup_{l \in N-\{l\}}(\text{eff}(l,S))))\text{;} \\
\beta &= \text{an estimate of the auxiliary time as a fraction of the machining time (we use } \beta = 0.2\text{).}
\end{aligned}
$$

7 Implementation

We have built a proof-of-concept implementation of these algorithms in C++ using version 1.5.1 of Spatial Technologies' ACIS© solid modeling system in conjunction with the NIH 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 MRSEV recognizer constructs instances of hole and pocket MRSEVs as outlined in [50, 51] with the exception some cases of through pockets. Implementation for general through pockets was restricted by the current version of the ACIS© application procedural interface which, at the time of this writing, we are extending to provide the needed functionality. The algorithms for building MRSEV models operate on any type of volumetric features.

8 Future Work

8.1 Recognizing MRSEVs

Near-term goals include incorporating a more sophisticated definition of accessibility, extending our results and implementation to include a wider class of MRSEVs, and exploring techniques to reduce computational costs.

8.2 Generating Redundant MRSEVs

If we use MRSEVs to represent the swept volume of the cutting portion of the tool, then we will need to take into account the possibility of using different tools when we generate alternative MRSEV models. For example:

1. It is often desirable to use a roughing operation to remove a volume of material followed by a finishing operation in which the swept volume of the tool completely subsumes the removal volume of the roughing operation. Examples are (i) making a hole by drilling and then reaming the hole and (ii) making a slot with a roughing end mill and then finishing the slot with a slightly larger finish end mill. It follows that redundant MRSEVs must be considered at some point. The redundant MRSEVs should certainly be generated before a cutting order is established and cost is estimated.

2. If we are cutting a pocket whose outline is an hourglass shape (or any shape with a bottleneck in it), the cost-effective method is to use a large tool to cut the bottom and top of the hourglass and a small tool to cut the narrow part in the middle where the large tool would not fit. Using the small tool to cut the entire pocket would take too much time. Thus, a MRSEV decomposition must include three MRSEVs for cutting the pocket. We are exploring techniques for identifying bottlenecks in a MRSEV, and splitting the the MRSEVs if bottlenecks occur.

3. For pocket MRSEVs, in some cases we assign an arbitrary tool radius. We are working on developing some heuristic rules to determine tool radius values when generating a MRSEV model.

8.3 Incorporating Setup and Fixturability Aspects in the Evaluation Framework

Our current approach does not deal with considerations involved with set up and fixturing issues. When evaluating MRSEV models (see Section 6), we need to make sure that all intermediate workpiece shapes can be clamped. Addressing this issue is a major problem for future work.

9 Conclusions

We have described our work toward the goal of developing a general approach to integrating CAD and CAM applications for the manufacture of machined parts. Our approach involves taking a CAD model, extracting alternative MRSEV models for that CAD model, and evaluating the MRSEV models to determine which one is optimal for the particular CAM application at hand. 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 are often different enough that integrating them can prove difficult. To address this problem, our approach encompasses many parts of direct interest to machining and manufacturability evaluation application and employs the MRSEV feature library, offering the possibility of compliance with the well known STEP standard.

2. Our approach to recognizing MRSEVs is complete over a significant class of realistic parts, even if the features intersect with each other [51]. Knowing the limits on completeness is useful in domains such as manufacturability evaluation, in which estimating the manufacturability of a design may require trying many alternative MRSEV models to find which is best.

3. Our approach handles hole, pocket, and through-pocket MRSEVs and their associated accessibility constraints. These MRSEV features represent a variety of milled shapes such as slots, steps, profiles, and slabs. The criteria for evaluating the alternative MRSEV models may be specified by the user, in order to satisfy the objectives of the user's particular CAM application. Potential CAM applications for our approach include process planning, NC part programming, fixture design and selection, and manufacturability evaluation.

Acknowledgements

This work was supported in part by NSF Grants NSFD CDR-88003012, IRI-8907890, and DDM-9201779, and by the University General Research Board.

References

[1] 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.

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

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

[4] A. Klien. A solid groove feature based programming of parts. *Mechanical Engineering*, March 1988.

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

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

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

[8] 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.

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

[10] Thomas J. Peters. Mechanical design heuristics to reduce the combinatorial complexity for feature recognition. *Research In Engineering Design*, 4:195–201, 1993.

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

[12] J. Corney and D. E. R. Clark. Face based feature recognition: Generalizing special cases. *International Journal of Computer Integrated Manufacturing*, 6(1 & 2):39–50, 1993.

[13] 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.

[14] Thomas R. Kramer. A parser that converts a boundary representation into a features representation. *International Journal of Computer Integrated Manufacturing*, 2(3):154–163, 1989.

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

[16] 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.

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

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

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

[20] Yong Se Kim and D. J. Wilde. A convergent convex decomposition of polyhedral objects. *Transactions of the ASME*, 114:468–476, September 1992.

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

[22] 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.

[23] 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.

[24] S. Meeran and M. J. Pratt. Automated feature recognition from 2d drawings. *Computer Aided Design*, 25(1):7–17, January 1993.

[25] 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.

[26] D. H. Lentz and R. Sowerby. Feature extraction of concave and convex regions and their intersections. *Computer Aided Design*, 25(7):421–437, July 1993.

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

[28] Tien-Chien Chang. *Expert Process Planning for Manufacturing.* Addison-Wesley Publishing Co., 1990.

[29] R. R. Karinthi and D. S. Nau. An algebraic approach to feature interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(4):469–484, April 1992.

[30] James G. Bralla, editor. *Handbook of Product Design for Manufacturing.* McGraw Hill Book Company, 1986.

[31] Thomas J. Drozda, editor. *Machining*, volume 1 of *Tool and Manufacturing Engineers Handbook.* Society of Manufacturing Engineers, fourth edition, 1983.

[32] H.P. Wang and J.K. Li. *Computer Aided Process Planning.* Elsevier Science Publishers, 1991.

[33] W. Winchell. *Realistic Cost Estimating for Manufacturing.* Society of Manufacturing Engineers, 1989.

[34] G. M. Zhang and S. C-Y. Lu. An expert system framework for economic evaluation of machining operation planning. *Journal of Operational Research Society*, 41(5):391–404, 1990.

[35] R. Bakerjian, editor. *Design for Manufacturability*, volume 6 of *Tool and Manufacturing Engineers Handbook.* Society of Manufacturing Engineers, 1992.

[36] R. Gadh, E.L. Gursoz, M.A. Hall, F.B. Prinz, and A.M. Sudhalkar. Feature abstraction in a knowledge-based critique of design. *Manufacturing Review*, 4(2):115–125, 1991.

[37] G. Harhalakis, A. Kinsey, I. Minis, and H. Rathbun. Manufacturability evaluation of electronic products using group technology. In *NSF design and manufacturing systems grantees conference*, Charlotte, North Carolina, 1993.

[38] C.C. Jara-Almonte and S. Krishnamorrthi. Evaluation of the manufacturability of machined components using axiomatic design principles. In *NSF design and manufacturing systems grantees conference*, Austin, TX, 1991.

[39] S. R. Shanker and D. G. Jansson. A generalized methodology for evaluating manufacturability. In W. G. Sullivan and H. R. Parsaei, editors, *Concurrent Engineering, Contemporary Issues and Modern Design Tools*, pages 248–263. Chapman and Hall, 1993.

[40] Aristides A. G. Requicha. Representation for rigid solids: Theory, methods, and systems. *Computing Surveys*, 12(4):437–464, December 1980.

[41] D. Schenck. Exchange of product model data – part 11: The EXPRESS language. Technical Report ISO TC184/SC4 Document N64, International Organization for Standardization, 1990. July.

[42] 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.

[43] J. Altemueller. The STEP file structure. Technical Report ISO TC184/SC4 Document N279, International Organization for Standardization, 1988. September.

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

[45] J. Van Maanen. Product data representation and exchange – part 21: Clear text encoding of the exchange structure. Technical Report ISO TC184/SC4 Document N78, International Organization for Standardization, 1991. March.

[46] Thomas R. Kramer. Issues regarding material removal shape element volumes (MRSEVs). Technical Report NISTIR 4804, The National Institute of Standards and Technology, Gaithersburg, MD 20899, March 1992.

[47] Michael J. Pratt. Tutorial paper on advanced topics in solid modeling: Form features and their application in solid modeling. In *SIGGRAPH 1987*. The Association for Computing Machinery, ACM Press, 1987.

[48] 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.

[49] Christoph M. Hoffman. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers Incorporated, CA, 1989.

[50] 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.

[51] 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.

[52] T. H. Corman, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw Hill, 1990.

[53] 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.

[54] Yun Peng and James A. Reggia. Diagnostic problem-solving with causal chaining. *International Journal of Intelligent Systems*, 2:265–302, 1987.

[55] D. S. Nau, G. Zhang, and S. K. Gupta. Generation and evaluation of alternative operation sequences. In A. R. Thangaraj, A. Bagchi, M. Ajanappa, and D. K. Anand, editors, *Quality Assurance through Integration of Manufacturing Processes and Systems, ASME Winter Annual Meeting*, volume PED-Vol. 56, pages 93–108, November 1992.

[56] S.K. Gupta and D.S. Nau. A systematic approach for analyzing the manufacturability of machined parts. *Computer Aided Design*, 1994.

[57] C. C. Hayes and P. Wright. Automatic process planning: using feature interaction to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[58] Machinability Data Center. *Machining Data Handbook*. Metcut Research Associates, Cincinnati, Ohio, third edition, 1980.

[59] F.W. Wilson and P.D. Harvey. *Manufacturing Planning and Estimating Handbook*. McGraw Hill Book Company, 1963.

[60] G. Zhang, D. Nau, W. Ko, and S. Gupta. Economic evaluation of alternative machining operation plans. In *NSF Design and Manufacturing Grantees Meeting*, January 1994.