Qualitative spatial reasoning for manufacturing features

Erik J. Johnson
erik@ece.arizona.edu

Michael M. Marefat
marefat@ece.arizona.edu

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

Abstract

One weakness within computer integrated manufacturing is an inability to effectively and efficiently model, reason, and make inferences about three-dimensional space. In this paper, we propose a qualitative model of a component according to its features to overcome this weakness. This model consists of two key ideas: *(i)* the model treats features qualitatively *i.e.* independent of size, orientation, and global position, and *(ii)* the model mathematically represents the spatial relationships among the features. The advantage of such a model will be its ability to reason about space in order to solve problems within the manufacturing processes. In addition to presenting the model, in this paper we will provide an example reasoning task of the model to find the multiple interpretations of a component.

keywords: spatial reasoning, qualitative reasoning, multiple interpretations, features

Introduction

Recently the integration of intelligence into the manufacturing environment has become increasingly important. The need for increased productivity and flexibility has led to advanced computer-aided design (CAD) tools and intelligent computer-aided process planning (CAPP) tools. From all of these needs, research has begun to focus on computer-integrated manufacturing (CIM) tools, which have proven successful in making the manufacturing processes integrated, intelligent, and flexible.

We have identified one common weakness with the CIM processes: an inability to effectively and efficiently model, reason, and make inferences about three-dimensional (3D) space. We propose a qualitative model of a component based on its features to overcome this weakness. This model consists of two key ideas:

- the model treats features qualitatively *i.e.* independent of size, orientation, and global position, and
- the model mathematically represents the spatial relationships among the features.

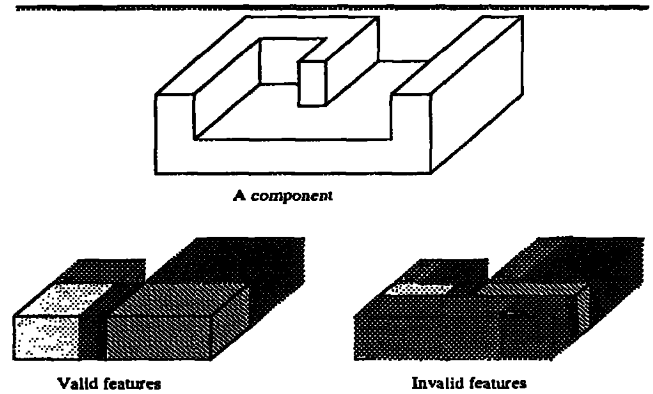


Figure 1: An example component from the manufacturing domain and a typical (valid) decomposition of the component's Δ volume into features. The bottom right shows similar features, but the decomposition is not considered intuitive (or valid). The difference between the two decompositions is based on the spatial locations of the features and their interactions.

As an example of this weakness, consider the simple component in Figure 1. Many of the planning tasks (process, inspection, assembly) within the CIM processes work with a description of the component based on a set of *features*. However, the input to the entire process is often a solid model (geometric description) and therefore needs translation into an equivalent set of features. At the heart of this translation process (often called *feature recognition*) is the need to reason about the spatial relationships amongst the faces of the component. Figure 1 shows several of the features within a component, and several features that do not intuitively belong within the component. The key to distinguishing the difference between the two types of features lies in spatially reasoning about the relative location and interactions of the faces in the component. To foreshadow the paper, Figure 4 shows our qualitative model for this component. Notice that both the features (vertices in the graph) and spatial relationships between the features (edges) are stored.

Finally notice that the problem of describing a component according to its features is ambiguous. It is possible (and likely) that there will be several valid *interpretations* of a component according to its features. Given the features within a component, the multiple interpretations can be found by testing for spatial interactions amongst the features. Our model will support reasoning tasks such as finding multiple interpretations. Figure 5 shows the multiple interpretations found for the component in Figure 1. These interpretations are directly extracted by searching our model for certain spatial relationships.

The contributions of the this work are summarized as:

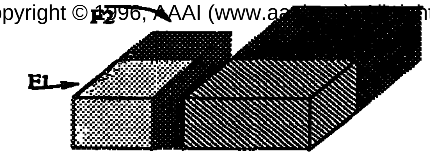
- the introduction of a qualitative model of a component based on its features,
- development of mathematical definitions (a systematic extensible vocabulary) for spatial relationships between two features, and
- an application of this model for describing the necessary steps in finding multiple interpretations and feature extraction.

Feature model and general definitions

In this section we provide a brief treatment of what we mean by a feature. This requires several notions pertaining to the half-spaces within a component as well as linear sweeps of 2D lamina. Due to space constraints, only the essential definitions are provided, however they will suffice to understand the rest of the ideas in this paper. For the scope of this paper, we will consider the common manufacturing features: prismatic pockets and holes, through and blind steps, through and blind slots, and cylindrical holes and pockets. However, the methodology is not strictly dependent upon such features. In the conclusion we will discuss the extension of this paper to arbitrarily defined linear sweep features. For the scope of the objects that we consider in this paper, we require that they have only planar and round (*i.e.* cylindrical) faces, and although these faces do not need to interact in a strictly orthogonal (or parallel) nature, we do require the object to be well behaved. That is, no dangling faces, disconnected objects, etc. Finally, for practical purposes, since we do not check for the manufacturability of the object, we restrict the objects to those which can be manufactured, although this is not a theoretical limitation.

We briefly digress to describe the input to, and the assumptions about, our system. As input to our methodology we require a solid model, and in particular we will use the notion of a boundary representation (Brep) for our solid model. Furthermore, we assume that we have the stock that the component came from, which effectively provides us with the delta volume (Δ volume) of the component. We represent features through linear sweeps[1] similar to (Gupta et al. 1993; Nau et al., 1993). We do so because a growing standard (*STEP* and *MRSEVs*) for the representation of features in manufacturing (Kramer, 1992) has its basis in swept features.

[1] Or just sweeps



Feature F1:
Type: "Blind slot"
Sweep face: vertices–((0,1,1), (–2,1,1), (–2,1,3), (0,1,3))
Sweep direction: (0, 1, 0)
Sweep interval: [0, 2]

Feature F2:
Type: "Pocket"
Sweep face: vertices–((–2,1,1), (–2,3,1), (–3,3,1), (–3,1,1))
Sweep direction: (0,0,1)
Sweep interval: [0,2]

Figure 2: Two example representations for swept features from Figure 1.

We define a feature as:

Definition 1 *A* **feature** *is a 4-tuple, (T, F, D, I), where T is the* type *or* label *(e.g. slot), F is the sweep face, D is the sweep direction (either parallel or antiparallel to the sweep face normal) and I is the (closed) sweep interval.*

In Figure 2, two of the features from Figure 1 are shown along with their symbolic representations.

One problem with representing features through sweeps is that we can create an infinite number of features simply by varying the sweep intervals. This of course is undesirable, so we define families of features which allow us to take an infinite number of features, and talk about them as simply one group of features.

Definition 2 *A* **family of features** *is a set of features all with the same label (T) and sweep face (F).*[2]

We need only one more essential definition and that is of a *canonical feature.* Our motivation for defining a family of features was to eliminate the possibility of having an infinite number of features within a component. Notice however, that a family of features only meets half of the definition for a feature (*i.e.* only the label (T) and sweep face (F)). A canonical feature will therefore add on the direction (D) and interval (I) to a family of features. It will be canonical features that we desire to extract and model.

Before we define a canonical feature, we must informally define a *partitioning.* A partitioning of a Δ volume is similar to cutting the Δ volume according to its half-spaces (see (Tseng & Joshi, 1994)). As an example, see Figure 3 (a). These cuts will form both

[2] Two sweep faces are the "same" if, for some length d, one face translated along its normal a distance of d is equal to the second face.

the faces of the canonical features and the intervals for the sweep lengths. Finally then:

Definition 3 *For a given component, let* $P = \{f_1, f_2, \cdots, f_n\}$ *be a partitioning of the* Δ *volume. Then a family of features with sweep face* F*, along with a sweep direction* D *and a given sweep length interval,* $[L_1, L_2]$*, define a* **canonical feature** *if:*

1. F *is the* same *as* f_i *for some* $1 \leq i \leq n$*, and*
2. $F + DL_1$[3] *and* $F + DL_2$ *are both subfaces of some support face*[4] *of the* Δ *volume.*

All of the features we will show throughout the rest of this paper meet the definition of a canonical feature. The importance of spending time to define such features is so that we can limit the number of features within a component to a finite number.

Building the qualitative feature model

Our qualitative feature model (QFM) is derived based on the metric diagram / place vocabulary (MD/PV) concepts of (Forbus, Nielsen, & Faltings, 1991). Because symbolically (*i.e.* qualitatively) reasoning about 3D space is often computationally expensive, the MD/PV model combines both quantitative knowledge (the MD) and qualitative knowledge (the PV) in an effort to ease the computational burden without sacrificing the qualitative inferencing power. The quantitative model, the MD, builds directly off of the input to our system, (*i.e.* a solid model), and provides a knowledge source which is queried for the construction of the place vocabulary. The place vocabulary is a completely qualitative model of the component, building directly from the metric diagram.

This section will describe the construction of our QFM in two stages: *(i)* First, the qualitative model without spatial relationships is described. *(ii)* Second, spatial relationships are defined.

In order to foreshadow this section, the overall construction of the QFM for a component is given as follows:

1. Partition the faces of the Δ volume of the component into sweep faces according to the Δ volume's half spaces.
2. *for each* pair of distinct sweep faces within the Δ volume *do:*
   2.1 Construct the metric diagram for these two sweep faces by forming a mathematical plot describing their intersection
3. *for each* metric diagram *do:*
   3.1 Query the MD for the intervals in which the sweep faces do and do not intersect
   3.2 *for each* of these intervals *do:*
   3.2.1 Form new features from these sweep faces and sweep intervals, and add these features into the place vocabulary
   3.2.2 Use the half spaces of these features to derive their qualitative spatial relationships
   3.3.3 Add the derived qualitative spatial relationships into the place vocabulary

[3]The notation $F + DL$ is defined as the new face, F' found by translating the face F by distance L in direction D.

[4]The support faces of the Δ volume of a component are defined by the half spaces of the Δ volume.

Metric diagrams

We begin then with the general characteristics of a metric diagram. The metric diagram should be built directly from the input to the system, (*i.e.* the solid model), and begins the conversion from the strictly quantitative input to the qualitative place vocabulary model.

Several MDs will be constructed for any given component and each MD will elicit information about families of features by providing the sweep intervals for which different families exist. The net effect is that the MDs will *(i)* provide a set of families of features sufficient to find all canonical features for a given component, and *(ii)* provide interaction information for each pair of intersecting feature families.

Definition 4 *A* **metric diagram** *between two linear sweeps is a 2D-plot such that:*

1. *The axes each represent the length of one of the sweeps. A negative value represents a sweep length in the opposite direction of the sweep direction.*
2. *All points of intersection between the two leading faces of the sweeps are on the plot.*
3. *The maximum (minimum) value of either axis is the length at which the forward (reverse) sweep leaves the stock or intersects the part.*

Figure 3 (b) shows several example MDs. To interpret these diagrams, take for example the first MD between the faces p_1 and p_3. Each tick on the graph indicates one unit of length. The solid dark line indicates the intersection points of the faces p_1 and p_3 swept at a given length. For example, while p_1 is in the interval $[0, 2]$ and p_3 is in the interval $[1, 3]$ the faces are intersecting.

To understand the significance of the dashed lines shown on many of the MDs, we observe the following:

Observation 1 *If either axis in a MD is divided into intervals based on the first and last points of intersection, these intervals correspond to the intervals of canonical features.*

This observation is one of the most important uses of the MDs. The dashed lines indicate the first and last points of intersection. Again referring to Figure 3 (b) and (c), notice that if we sweep face p_1 in the interval $[0, 2]$ we form the canonical feature F_1. Furthermore, if we sweep p_1 in the intervals $[0, 5]$ and $[2, 5]$ we form

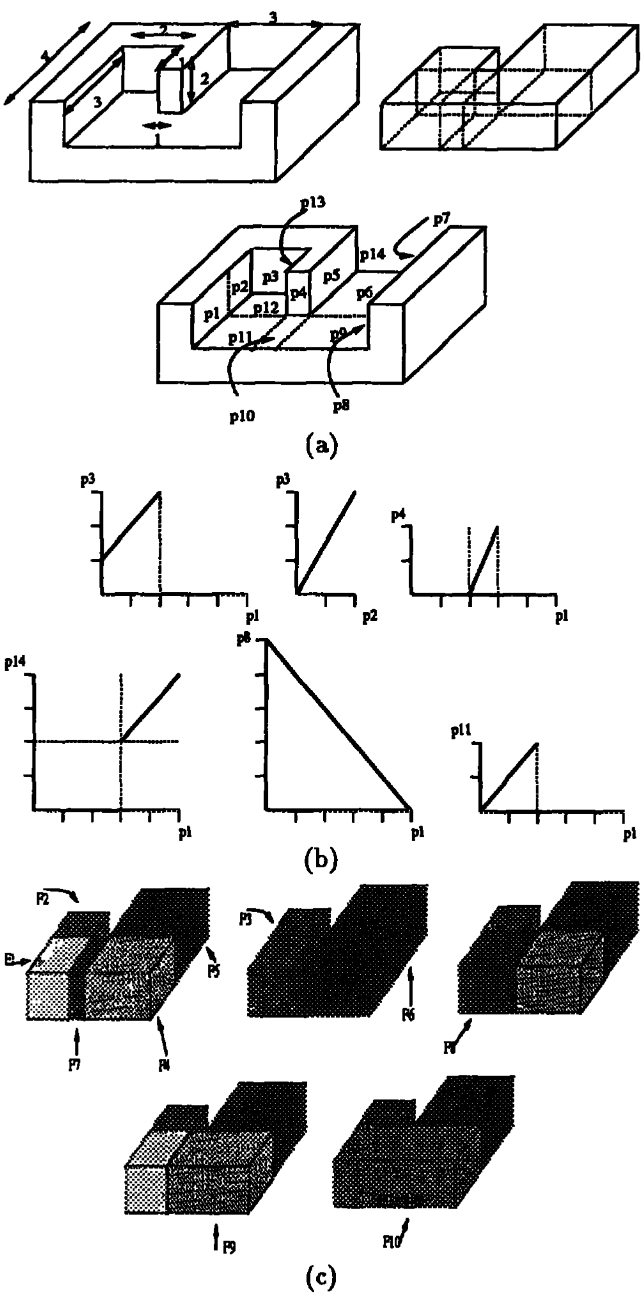


Figure 3: (a) Shows a component along with the cut faces of its Δ volume. (b) Shows several of the MDs created for the component. (c) Shows the canonical features found for the component.

Cut faces:

1. Let $F = \{f_1, f_2, \cdots, f_n\}$ be the set of planar faces of the Δ volume, p_i be the support plane of f_i, $P = \{p_1, p_2, \cdots, p_n\}$, and, $E' = \emptyset$.
2. *for each* $p_i \in P$ *do:*
   // Build the support face for each face in the Δ volume
   3. $s =$ **Support face** (p_i, i, F)
   // Use this support face to cut the other faces of the Δ volume
   4. *for each* $f_j \in F$ *do:*
      5. *if* (s is not parallel to f_j) *then:*
         6. $e' =$ the edge of intersection of s and f_j
         7. $E' = E' \cup \{(e', j)\}$
9. **Split faces** (F, E')

Support face (p, i, F)
1. $E = \emptyset$
// Compute the edges of the maximal support face
2. *for each* $f_j \in F$ *do:*
   3. *if* (f_j is not parallel to p) *then:*
      4. $e =$ the edge of intersection between f_j and p
      5. $E = E \cup e$
// Compute the maximal support face
6. $m =$ the face formed by p and E
// Decompose the maximal support face into maximal convex faces
7. $C =$ **Maximal convex decomposition**(m)
// Eliminate convex faces that do not intersect the original face
8. *for each* $f \in C$ *do:*
   9. *if* (f does not intersect the interior of f_i) *then:*
      10. $C = C - f$
// Union of the remaining convex faces is the support face
11. *return* $\bigcup C_k$

Algorithm 1: Cutting the faces of the Δ volume.

the features F_{10} and F_9 respectively. Clearly the motivation for constructing MDs is to extract the canonical features in this manner.

In the remaining portion of this section, an algorithm for the construction of the MDs is provided and discussed. Two questions beg attention with respect to the construction of the MDs: *(i)* First, what faces of the object will be swept? *(ii)* Second, what different sweep lengths should be checked to see if two sweeps intersect or not?

Finding the faces to sweep - (partitioning): To answer the first question of what faces to sweep, two issues are addressed: *(i)* enough faces should be swept to insure a complete model. Specifically, these sweeps should elicit a sufficient number of families of features to find all canonical features. *(ii)* A systematic approach should be taken to avoid ad-hoc, yet appealing, methods to find such faces.

Construct MD

1. Let S be the set of support faces for a part;
   let f_1 and f_2 be two faces from F in **Cut Faces**.
2. For each support face compute the distance between it and one of the two sweeps in the MD
3. Compute the distance between the other sweep in the MD and every other support face
4. See if the two sweeps intersect at each of these distances
5. If so, save this information into the MD.

Algorithm 2: Finding the important sweep lengths in a MD.

The solution is to cut the faces of the Δ volume (similar to (Dave & Sakurai, 1995; Tseng & Joshi, 1994)) according to the half-spaces of the component. Algorithm 1 shows the steps for finding the faces to sweep, and Figure 3 (a) shows such a partitioning (or cutting).

Finding the sweep lengths: The second question for the MDs is what sweep lengths will be considered important? Recall that each MD determines at what points its two sweeps do and do not intersect. It is not very intelligent, or practical, to simply start testing every sweep length based on some small interval. Not only would the algorithm be considerably expensive, we run the risk of missing an important sweep length. The solution is to use the same planes that cut the Δ volume to produce potential sweep lengths to examine. Algorithm 2 shows how to find these important sweep lengths and build a single MD for two given faces.

We conclude this section by summarizing the important points about the MDs:

- A MD is a mathematical plot relating two sweep faces
- MDs elicit canonical features from the solid model of the component. They do so by:
  - Partitioning the faces of the Δ volume according to the support faces in the Δ volume
  - Using this partitioning along with the support faces to construct plots (the actual MDs) between pairs of sweep faces

The place vocabulary

In this section we describe the qualitative half of the MD/PV model, the place vocabulary. The PV is our qualitative feature model (QFM), however in this section we neglect the spatial relationships that will be incorporated and described in the next section.

The construction of the PV should involve the querying of the MDs for spatial information. The underlying goal is to move away from the actual size of the features, and represent the features and their interactions in a qualitative (symbolic) format.

The PV is a labeled graph, in which each edge is augmented with additional information (*i.e.* its label). The PV accomplishes two tasks: *(i)* First, it represents the features of the component qualitatively. The features will no longer be dependent upon actual size or orientation. *(ii)* Second, the PV will record which features interact (*i.e* intersect), and what spatial relationship exists between each pair of interacting features.

Definition 5 *The* **place vocabulary** *for a given component is a labeled graph whose vertex set, V, consists of every canonical feature in the component, and two vertices are adjacent if and only if the corresponding canonical features intersect. Each edge is labeled with two properties:*

1. *The type of intersection of the two features incident upon the edge; the type is either* subsumes, internal, face *or* end.
2. *The qualitative spatial relationship of the two features incident upon the edge.*

For two canonical features, the intersection type *subsumes* intuitively means that one feature completely subsumes the other; the intersection type *internal* means that the two features only intersect strictly in the interior of their sweep length intervals; the intersection type *end* means that at least one of the two features intersects the other at the start or end of its sweep length interval, and the intersection type *face* means the two features only intersect at a face. Note these are not spatial relationships, rather they are defined volumetric and interval comparisons. These are defined as follows:

Given two features, F_i and F_j, with sweep length intervals $[s_i, e_i]$ and $[s_j, e_j]$ respectively. Let the interval of intersection in their MD be $[a, b]$, then the spatial relationship, $\mathcal{R}$, where $F_i \mathcal{R} F_j$, is of type:

subsumes if F_i and F_j are in the same family and $[s_i, e_i] \supseteq [s_j, e_j]$ or $[s_j, e_j] \supseteq [s_i, e_i]$

internal if F_i and F_j are not in the same family and $[s_i, e_i] \supset [a, b]$, $[s_j, e_j] \supset [a, b]$, and $s_i \neq e_i \neq a \neq b$, $s_j \neq e_j \neq a \neq b$.

end if F_i and F_j are not in the same family and $[s_i, e_i] \supset [a, b]$, $[s_j, e_j] \supset [a, b]$, and either $s_i = a$ or $e_i = b$ and either $s_j = a$ or $e_j = b$.

face if F_i and F_j intersect exactly at a face.

The PV is straightforward to compute given the MDs. Algorithm 3 shows the outline for computing the PV for a component. The first element of this algorithm is to query the MDs for the sweep faces found and the intervals created by the intervals of intersection. These two pieces of information provide the basis for the features within the component, and these features become vertices within the PV. Note that the MDs are redundant, that is, there could be several sweep faces and intervals that create the same feature. This situation is easily recognized however, since the

Build PV:

1. Let $S = \{$all of the sweeps found in the MD generation$\}$,
2. For each sweep in the MDs, compute all of the canonical features associated with it
   3. Divide the sweep lengths for this sweep according to the support planes of the Δ volume.
   4. For each interval defined by these divisions
      5. If the feature found is not in V then add it to V
6. For each pair of vertices, construct an edge between them if they intersect and add this edge to E.
7. V is the node set of the place vocabulary, and E the labeled edges.

Algorithm 3: Constructing the place vocabulary.

intervals of these faces must be the intervals of intersection within the MD. For example, in Figure 3, the sweep face p_1 in the interval $[0, 2]$, and the sweep face p_3 in the interval $[1, 3]$ both create the feature F_1. But these intervals are the intervals of intersection on the MD between p_1 and p_3. Since we do not want to add F_1 into our PV twice, we must check to see that two sweep faces and intervals do not create the same feature. The fact that the intervals in question ($[0, 2]$ and $[1, 3]$) are the intervals of intersection, tells us that it is possible these sweep faces and intervals create the same feature. If the two intervals were not the intervals of intersection, it would not be possible for the features to be the same.

The second element of the algorithm is to compute the edges within the PV. These edges come about from interactions of features. Again we utilize the MDs to tell us if an edge should be formed between two vertices (features). If the intervals that form two features overlap the interval of intersection on their corresponding MD, then there should be an edge created. For example, the features F_1 and F_3 in Figure 3 (c) clearly interact. The feature F_1 can be formed from p_1 in the interval $[0, 2]$. The feature F_3 can be formed from p_3 in the interval $[0, 3]$. These intervals both overlap the interval of intersection on the MD between p_1 and p_3 and therefore we know to create an edge between them.

Finally Figure 4 shows the PV for our example component. We have limited the edges to those of type *subsumes* or *end* because it simplifies the figure, but also because these are the only two edges of consequence when constructing the multiple interpretations of the component. The edges of type *face* and *internal* are important for other reasoning tasks not presented in this paper. The spatial relationship shown in the figure will be dealt with next.

In conclusion, some important properties of the PV are:

- The PV for a component is a graph whose vertices are canonical features, and edges represent spatial interactions between two canonical features.
- Each vertex is a canonical feature, identified by its label. The specific sweep information for this feature is maintained within the MDs, and hence the PV is independent of size of features.
- Each edge represents a spatial relationship between two canonical features. These spatial relationships are relative (see the next section) and hence the PV is independent of orientation.

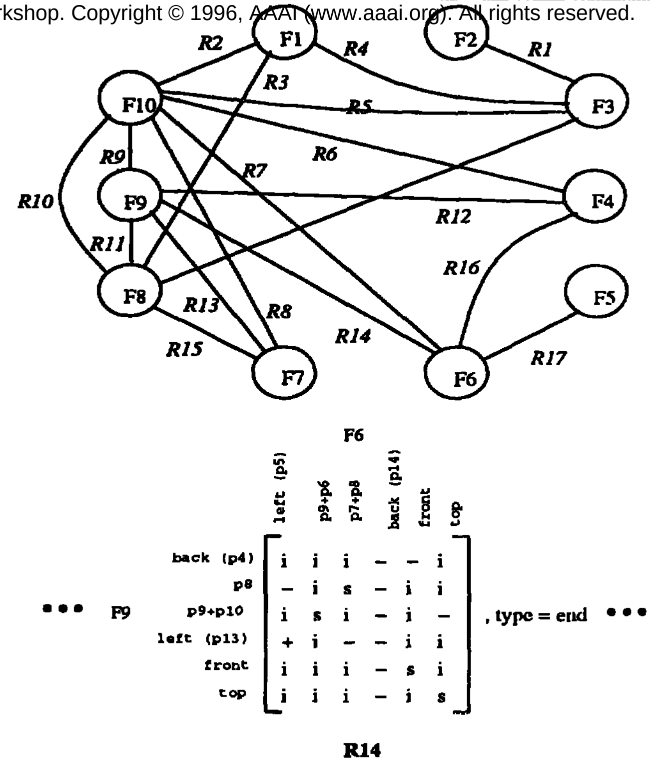


Figure 4: The place vocabulary for the example component. Only the edges of type *subsumes* or *end* are shown for simplicity.

A qualitative spatial vocabulary

In this section we define and describe a mathematical representation for the spatial relationship between two features. Spatial relationships are crucial to our notion of a qualitative feature model (QFM) and they should possess the following properties:

- The relationship should be relative to the features in question, not fixed to a global coordinate frame. This gives our QFM (or PV), the property of independence from orientation.
- The relationship should be qualitative, not so rigid that local variations in feature size or position effect it.

In addition, we also must avoid ad-hoc and ambiguous descriptions of spatial relationships such as "to the left of," and "perpendicular to."

The proposed spatial relationships are motivated by the works of (Allen & Koomen, 1983; Allen, 1983) and an extension of this work (Mukerjee & Joe, 1990). The notion of a half space (Freeman, 1990) is also integral to the relationship. Given a face, consider the plane passing through this face, called the *support plane*. The support plane divides space into two half spaces. The half space in which the normal of the face points is dubbed the *positive* half space (denoted $f+$) and the other the *negative* half space (denoted $f-$).

Definition 6 *Given two features, F_1 and F_2, such that F_1 is bounded by the set of faces $\{p^1_1, p^1_2, \cdots, p^1_n\}$, and F_2 is bounded by the set of faces $\{p^2_1, p^2_2, \cdots, p^2_m\}$*[5] *the* **qualitative spatial relationship**, *denoted $F_1 \mathcal{R} F_2$, is an $m \times n$ matrix of characters, where*

$$\mathcal{R}_{i,j} = \begin{cases} + & \textit{if } p^2_i \textit{ lies on the positive side of } p^1_j \\ - & \textit{if } p^2_i \textit{ lies on the negative side of } p^1_j \\ s & \textit{if } p^2_i \textit{ lies in the same plane as } p^1_j \\ i & \textit{if } p^2_i \textit{ intersects the plane of } p^1_j \end{cases}$$

.

Figure 4 shows the spatial relationship between F_6 and F_9. To interpret the matrix, the first column represents the position of each of the six faces of F_9 to the plane passing through p_5. The second column is similar, except for it is dealing with the plane passing through p_6. Also note that the edge between F_6 and F_9, labeled $\mathcal{R}_{14}$ is not only this spatial relationship, but it is also of type *end*. It should be apparent that F_6 and F_9 do indeed intersect at their "ends."

Finally now our model is complete. The important points to notice are that this definition has our desired properties:

- The use of half spaces makes the relationships invariant to local variations. Since we are only checking if a face is on one side of another face, it is not dependent on exact positioning of the faces.
- The relationship, $F_1 \mathcal{R} F_2$, is *intrinsic* (Retz-Schmidt, 1988) to F_1, which makes the entire relationship relative, not dependent upon global orientation.

Generating multiple interpretations

Researchers have become aware of the value of multiple interpretations (Gupta, Regli, & Nau, 1994; Gupta et al. 1994; Karinithi & Nau, 1992; Karinithi, Nau, & Yang, 1992; Nau et al., 1993; Regli, Gupta, & Nau, 1994; Tseng & Joshi, 1994) because they provide planners with the opportunity to explore alternate plans, and hence, find optimal plans. In this section we describe a systematic mechanism for the generation of the multiple interpretations of a component using our QFM. The proposed mechanism utilizes the PV, along with graph theory, and the notion of *envisionments* (de

[5] For an object with round faces, we simply place a bounding box around these faces.

Construct envisionment:

1. Let PV = the place vocabulary for a component.
2. Mark all vertices in PV as not yet in an interpretation
3. For each vertex in PV not yet in an interpretation
   4. Find all maximal independent sets of nodes containing this vertex in PV.
   5. Add each of these sets as a root to leaf path in the envisionment
   6. Mark every node in the envisionment as in an interpretation

Algorithm 4: Construct the envisionment for a component by finding maximal independent vertex sets from the PV.

Kleer, 1977) from qualitative reasoning. The overall mechanism will be to construct an envisionment from the PV, where the envisionment will represent the interpretations of a component by storing both canonical features and connections between sets of canonical features that constitute different interpretations.

To begin, envisionments have been used in qualitative reasoning to represent all possible feasible states of a solution, and possible paths through these states which lead to a solution of the problem. For our problem, in an intuitive sense, we use envisionments to represent the different interpretations of a component. The states of our envisionment are the canonical features, and paths through these states represent sets of canonical features; these sets of canonical features correspond to interpretations.

Definition 7 *An* **envisionment** *for a component is an acyclic graph, whose vertices are canonical features, and there is at least one vertex which is dubbed a "root," and one vertex dubbed a "leaf," and which has the following properties:*

- *Any path from any root vertex to any leaf vertex represents an interpretation of the component. That is, the vertices along such a path represent canonical features, and these features collectively represent an interpretation of the component.*
- *All paths from root vertices to leaf vertices collectively represent all possible interpretations of the component.*

The algorithm for generating envisionments is shown in Algorithm 4. The key to the algorithm is the notion of maximal independent sets of vertices within the PV. An *independent* set of vertices in a graph is a set of vertices that are all pairwise non-adjacent. A *maximal independent* set of vertices is an independent set such that no other independent set strictly subsumes it. In essence, the notion of independent sets of vertices in the PV systematically captures our notion of what canonical features do and do not belong together in an interpretation.

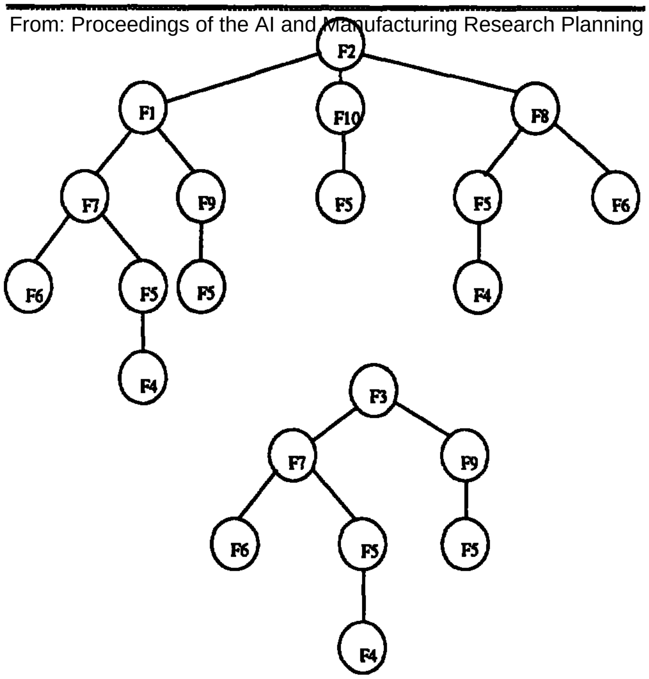


Figure 5: The envisionment for the previous component. Verify that each path from root to leaf is a maximal independent set of vertices in the PV given previously and is also an interpretation of the component.

Figure 5 shows the envisionment generated for our example component. Since the PV is relatively small, it should be easy to verify that each root to leaf path in the envisionment represents an interpretation of the component, but more importantly, is also a maximal independent set of vertices within the PV. For example, the path $\{F_2, F_1, F_7, F_5, F_4\}$ is an interpretation shown in Figure 3 (c), and the vertices $\{F_2, F_1, F_7, F_5, F_4\}$ in the PV are all pairwise non-adjacent. Furthermore attempting to add any other vertex into this set will result in some two vertices being adjacent.

Related works

Qualitative reasoning

Forbus (Forbus, Nielsen, & Faltings, 1991) developed the MD/PV model for its application in qualitative spatial reasoning. His application of the MD/PV model was to analyze, and determine the kinematics of, a clock. His motivation for proposing a model based on quantitative and qualitative information was his *poverty conjecture*, in which he argues that there is no completely qualitative model for 3D space. Our approach shares the generic concept of the MD/PV model from this previous work, however the approach, the formulations, and the developed MD/PV methods are quite different due to the different problem domains.

de Kleer (de Kleer, 1977) developed envisionments as one of the first notions in qualitative physics. His particular application of envisionments was to store the qualitative states of the position of a ball moving along a track. The envisionment allowed for the problem of the ball on the track to be analyzed, and automatically reasoned about, even without specific parameters about the scenario, like the initial velocity of the ball, etc. Again, although we share the concept of envisionments with de Kleer, our approach, our applications, and our developed envisionment techniques are for distinctly different problem domains.

Feature extraction and multiple interpretations

Several works (Gupta, Regli, & Nau, 1994; Gupta et al. 1993; Gupta et al. 1994; Nau et al., 1993; Regli, Gupta, & Nau, 1994) have recently been published relating to the topic of multiple interpretations and feature extraction. These works focus on both the completeness of the algorithms as well as the manufacturability of the interpretation. In this light, these works are not attempting to generate *all* interpretations, rather all interpretations that are both manufacturably feasible and not extremely expensive to manufacture.

Karinthi and Nau (Karinithi & Nau, 1992; Karinithi, Nau, & Yang, 1992) proposed a way to mathematically extract the multiple interpretations of a part given one interpretation. Given a set of features this system applied "algebraic" operators such as truncation and extension to try and form new features. These new features then could be used in a new interpretation. The exponential algorithm took combinations of features and applied the algebraic operators until it could not generate any new features.

Tseng and Joshi (Tseng & Joshi, 1994) approached the problem of feature extraction and multiple interpretations using a brute force algorithm. There were two steps involved, the first was to decompose the Δ volume into small pieces. The second step was to recombine these pieces into recognizable features. The recombination step enumerated all possible volumetric combinations.

Dave and Sakurai (Dave & Sakurai, 1995) presented a method for generating multiple interpretations based on the work of Tseng but their work addressed a few of the previous shortcomings. This work first decomposed the Δ volume into minimal cells, but then it went further to combine these minimal cells into maximal ones. These maximal volumes, which could possibly overlap, would then be the building blocks of the algorithm to find the different features and interpretations of the part.

Previous work in spatial relationships

Allen's (Allen & Koomen, 1983; Allen, 1983) work has previously been extended to include multi-dimensional domains by Mukerjee (Mukerjee & Joe, 1990). In this work, each dimension in the problem domain is represented by a one dimensional interval in a similar fashion to Allen. This work provides a basis for a three dimensional spatial model for describing the relative positions of two objects. The work can represent objects at arbitrary angles and positions, however is greatly simplified when all of the interactions are orthogonal. Although the spatial relationships we develop in this paper do not subsume those of Mukerjee, our spatial relationships are simplified and suited to the domain of features.

(Chen, Miller, & Lu, 1992) previously incorporated spatial relationships into the domain of features. In this work two types of spatial relationships were defined between features: *Is_In* and *Adjacent_To* (similar to our *subsumes* and *face* types). Using these spatial relationships, this research focused on analyzing the manufacturability of a given component by looking for feature interactions that created undesirable manufacturing situations. The relationships themselves were obtained by directly analyzing the solid model of the component.

Conclusion

In this paper we presented a framework and methodology for modeling a component according to its features. The presented method develops from the solid model of the component to a graph which qualitatively represents the component. Our model is qualitative in the sense that it is independent of the size of the features, the global orientation of the component, and local spatial variations in feature positions. Two key properties of our QFM are *(i)* features are treated qualitatively, and *(ii)* spatial relationships between features are maintained.

To achieve this task we employ the metric diagram / place vocabulary concepts which *(i)* provide a quantitative database to be queried about sweep faces and sweep intervals within the component (the MDs), and *(ii)* provide a qualitative model (the PV) linked to the quantitative data. This qualitative model can then be used to efficiently and effectively reason about the features within the component.

Furthermore, we have shown how to apply our QFM to reason about a component to find its multiple interpretations.

The contributions of the work can be seen as:

- A qualitative model of a component according to its features;
- the definition and incorporation of spatial relationships within the model, and
- the ability to reason with the model. Specifically, to find multiple interpretations.

We are currently pursuing several areas with respect to our QFM. First, we alluded to earlier that our model is not limited to a fixed set of features. It is possible to represent arbitrary features with our model so long as the features themselves are representable via linear sweeps. Our definition of spatial relationships already incorporates arbitrary feature definitions. Moreover, our PVs and envisionments do not need any modifications, however the construction of the MDs does require some modifications. We must now consider sweep faces that are not necessarily convex in nature. This can be accomplished by establishing ***acceptable*** and ***non-acceptable*** forms of MDs, along with a resolution strategy. When a non-acceptable MD is generated, the resolution strategy (often union, intersection, or difference) is applied to the two sweep faces in the MD, and a new sweep face is generated. Currently we are investigating what such a change does to the completeness and soundness of our model.

In addition to the above, we have also identified the following reasoning tasks which are possible with our QFM:

- Drawing analogies between components. Analogies allow re-use of previous knowledge by providing a means to index into object databases.
- Reasoning backward from the QFM towards the solid model. Such a task accomplishes two things: *(i)* it categorizes components into "families of components,", and *(ii)* it provides a good basis for critical evaluation of the qualitative nature of our model.
- Identifying symmetric interpretations. Due to our use of spatial relationships, interpretations of a component that are simply symmetric to each other can be detected, thus alleviating the burden from the planners.

References

Allen, J.F., Koomen, J.A., 1983. Planning Using a Temporal World Model, In *Proceedings of IJCAI-83*, Karlsruhe, West Germany: 741-747.

Allen, J.F., 1983. Maintaining Knowledge about Temporal Intervals, In *Communications of the ACM*, **26**(11): 832-843.

Chen, Y.-M., Miller, R.A., Lu, S.-C., 1992. Spatial Reaonsing On Form Feature Interactions For Manufacturing Assessment. In *Proceedings of ASME Conference on Computers in Engineering*, **1**: 29-36.

Dave, P., Sakurai H., 1995. Maximal Volume Decomposition and its Application to Feature Recognition, In *Proceedings of ASME Conference on Computers In Engineering*, Minneapolis, MN: 553-568.

de Kleer, J., 1977. Multiple Representations of Knowledge in a Mechanical Problem-Solver, In *Proceedings of the IJCAI*: 299-304.

Forbus, K.D., Nielsen, P., Faltings, B., 1991. Qualitative spatial reasoning: the CLOCK project, In *Artificial Intelligence* **51**: 417-471.

Freeman, H., 1990. *Machine vision for three-dimensional scenes*, Academic Press, Boston.

Gupta, S.K., Regli, W.C., Nau, D.S., 1994. Manufacturing Feature Instances: Which Ones to Recognize? Technical report ISR-TR-94-81, Institute for Systems Research, Computer Science Department at University of Maryland, College Park.

Gupta S.K., Kramer, T.R., Nau, D.S., Regli, W.C., Zhang, G., 1993. Building MRSEV Models for CAM Applications, Technical report ISR-TR-93-84, Institute for Systems Research, Computer Science Department at University of Maryland, College Park.

Gupta S.K., Nau, D.S., Regli, W.C., Zhang, G., 1994. A methodology for systematic generation and evaluation of alternative operation plans, In *Advances in Feature Based Manufacturing* Elsevier/North Holland, *eds.* Shah, J., Mantyla, M., Nau, D.: 161-184.

Karinthi, R.R., Nau D., 1992. An Algebraic Approach to Feature Interactions, In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(4): 469-484.

Karinthi, R., Nau, D., Yang, Q., 1992. Handling Feature Interactions in Process-Planning, In *Applied Artificial Intelligence* **6**: 389-415.

Kramer, T.R., 1992. A library of material removal shape element volumes (MRSEVs), Techincal report from The National Institute of Standards and Technology, NISTIR 4809, Gaithersburg, MD.

Mukerjee, A., Joe, G., 1990. A qualitative model for space, In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI)*: 721-727.

Nau, D., Gupta, S.K., Kramer, T.R., Regli, W.C., Zhang, G., 1993. Development of Machining Alternatives, based on MRSEVs, In *Proceedings of ASME Conference on Computers in Engineering* San Diego, California: 47-57.

Regli, W.C., Gupta, S.K., Nau, D.S., 1994. Extracting Alternative Machining Features: An Algorithmic Approach, Technical Report ISR-TR-94-55, Institute for Systems Research, Computer Science Department at Universityof Maryland, College Park.

Retz-Schmidt, G., 1988. Various Views on Spatial Prepositions, In *Artificial Intelligence Magazine*: 95-105.

Tseng, Y., Joshi, S., 1994. Recognizing multiple interpretations of interacting machining features, In *Computer-Aided Design* **26**(9): 667-688.