RECOGNIZE FEATURES FROM FREEHAND SKETCHES

Teng-Shang Hwang
Product Development Division
SDRC
Milford, Ohio

David G. Ullman
Department of Mechanical Engineering
Oregon State University
Corvallis, Oregon

ABSTRACT

A feature recognition system which allows the computer to capture sketches made by a mechanical designer is described. The system takes the user's input sketching actions and interprets the collection of 2D geometric elements to extract 3D information from them in terms of mechanical features. A application domain, knowledge representation, the derivation of feature recognition rules, inference mechanism, and conclusions are presented in this article.

1. INTRODUCTION

Mechanical CAD/CAM systems are much more sophisticated than they were a decay ago. Variational geometry, parametric modelling, and feature based design technologies couple with dynamic user interaction make solid modellers very easy to use and has been widely accepted by mechanical design engineers. However, current CAD/CAM systems are not smart enough, not intuitive enough for conceptual design.

With the development of expert system, human expertise is able to be coded with computer languages. The coded knowledge grants computers the capability to simulate the behaviour of human experts to solve specific problems in a particular domain of knowledge. The research of "The Design Capture System [Hwang 90, 90a]" is part of an effort in the research community of future CAD/CAM system development. This paper discusses one of three modules in the Design Capture System, a features recognition subsystem for recognizing three-dimensional mechanical form features from two-dimensional freehand sketches.

The capability of interpreting 2D sketches and extracting 3D data from them is the basis of creating a sketching tool that allows engineers to sketch their ideas on the computer in a natural way. In order to convert a designer's conceptions into a three-dimensional form on the computer, current CAD systems force designers to concentrate on modelling rather than on designing. This causes interruptions in the designer's decision-making process. "The greatest weakness of current CAD tools is their inability to represent mentally conceived forms in a natural manner. Designers are hampered in the design process by not being able to represent their ideas in a graphical language that is compatible with their thought process [Ullman 87]."

The 3D sketching tool will allow designers to concentrate on conceiving designs with high abstraction level features instead of concentrating on modelling objects with low abstraction level geometric primitives. This change will be an important step in the upgrading of the CAD system from a drafting and modelling tool to a design tool.

2. BACKGROUND

The extraction of three-dimensional information from a single-view drawing has been studied for decays. Huffman [Huffman 71] and Clowes [Clowes 71] introduced the "Huffman-Clowes labelling method" to classify line segments in drawings into three categories: 1) convex edges formed by both side faces facing toward a viewpoint, 2) convex edges formed by one side surface facing the viewpoint and the other side surface facing opposite the viewpoint, 3) concave edges. This reduces line drawing interpretation problems to a problem of systematically assigning labels to edges of the drawing. This method has been verified in various kinds of drawings [Sanker 77] [Sugihara 78] [Kanade 81] [Lee 85]. One of the problems of the labelling scheme is that the interpretation is not unique. Several candidates could be generated from a single drawing. Many ideas have been proposed to precisely define the object with the labelling scheme: reciprocals in a gradient space [Huffman 78], a figure-construction approach [Shapira 85], and Linear algebra representation [Sugihara 84, 86]. Pugh [Pugh 89] proposed an algorithm which applies geometric constraints satisfaction to the labelling scheme in order to precisely define the object by allowing users to place geometric constrains on the line drawing.

3. THE DESIGN CAPTURE SYSTEM

The design capture system consists of three subsystems: a 2D freehand sketching subsystem [Fang 88], a features recognition subsystem, and a spatial reasoning subsystem. It operates in the following manner. The freehand sketching subsystem allows designers to sketch their ideas onto the computer using a stylus with a tablet, in the same manner as using a pencil and paper. (Figure 1a). The freehand sketch is recognized on-the-fly as one of six 2D geometric elements such as line, circle, ellipse, etc. and is displayed on the screen "cleaned-up", as shown in Figure 1b. The feature recognition subsystem interprets the collection of 2D geometric elements in order to extract three-dimensional descriptions from them and then constructs a mechanical feature, as shown in Figure 2. The sketched three line segments in Figure 2a define the length, the width, and the height of a rectangular solid in the isometric manner which are then interpreted and displayed as a feature (block) in Figure 2b. The spatial reasoning subsystem finds relationships between a "being added" feature and the existing object. Figure 3a shows an ellipse and a vertical line sketched on the interpreted block. The newly added sketch is interpreted as a cylinder by the features recognition subsystem (Figure 3b). The spatial reasoning subsystem then infers that the cylinder is located at the top face of the block.

This paper is devoted to the design and implementation of the second subsystem, the feature recognition subsystem.

4. THE FEATURE RECOGNITION SYSTEM

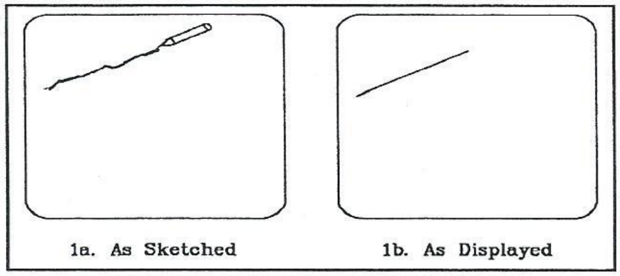

FIGURE 1. SKETCH INTERPRETATION

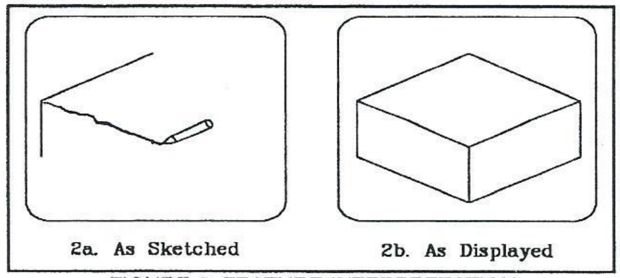

FIGURE 2. FEATURE INTERPRETATION

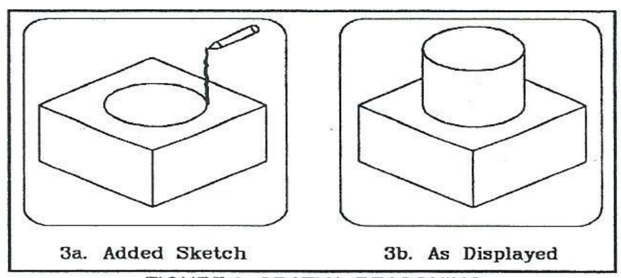

FIGURE 3. SPATIAL REASONING

Representing designs solely in terms of solid models is inadequate for an intelligent CAD/CAM system. A general consensus on the need to represent aspects of design on a more abstract level has been reached in the automated design and manufacturing research community [Wentorf 89]. It is believed that the design process will be facilitated by creating designs in terms of features and that the application of automated design and manufacturing will be made possible or will be improved significantly by the use of the data provided by the use of the feature model. This belief is currently being investigated by many researchers [Luby 86] [Vaghul 85] [Sakurai 88] [Henderson 88] [Shah 88,89] [Hwang 90, 90a]. They have built the feature model in two approaches. The first approach involves extracting feature information from the solid model; the second approach involves creating design in terms of features, i.e., design with features. The design with features approach was chosen for the development of the system because it has the following advantages.

1) It provides a domain specific features library which enables the designer to perform design in terms of features which are meaningful to designers and manufacturers.

2) Once the design is completed the feature model is available. This makes unnecessary the arduous task of decomposing the solid model in order to create feature models.

3) Many design rules and manufacturing processes are described in terms of features. This makes feature model a much highly desirable design representation for the automated environment. Here are some design rules for the domain of injection molded parts, which are described in terms of features:
   - -Provide *tapers* from the *split-line*.
   - -Aim at uniform *wall thicknesses*.
   - -Avoid *undercuts*.
   - -Enhance thin long *walls* with *ribs*.

4) It is suited for parametric design. One aspect of the design with features system is building the design feature by feature in temporal order. The temporal natural of the data developed this way sets up a natural dependency for parametric changes of the part.

In order to use features to create designs, the feature recognition subsystem has been developed to extract feature information from 2D freehand sketches. Here, an expert system is used for matching the combination of 2D graphic elements (line segments, ellipses, circles, etc.) to feature definitions to determine which of these features has been sketched. The details of the feature recognition subsystem are discussed below in terms of: application domain, knowledge representation, rules for feature recognition, and the inference mechanism.

4.1 Features To Be Recognized

The number of features required for constructing complex mechanical parts is huge. The problem of "features explosion" is one of the potential obstacles to implement the "design with features" approach [Vaghul 85]. To make the development of this prototype system possible, the application domain has been limited to plastic injection molded parts and only the most common features (see Table 1) were selected for implementing the system.

TABLE 1. FEATURES LIST

- block_structure
  - wall
  - rib
  - window
  - pocket
  - protrusion (rectangular)
  - depression (rectangular)
- cylindrical_structure
  - boss (solid)
  - boss (hollow)
  - disk
  - bulge
  - hole (blind)
  - hole (through)
  - protrusion (cylindrical)
  - depression (cylindrical)
- spherical_structure

4.2 Knowledge Representation

Webster's dictionary defines the word *knowledge* as "the fact or condition of knowing something with familiarity gained through experience or association". In the field of artificial intelligence, the term *knowledge* means the formatted information used by a computer program to behave intelligently. This information is extracted from human experts and is structured in a manner that makes the problem easier to solve. The technique for structuring information in an effective format is called *knowledge representation*. The most widely used techniques for knowledge representation in current expert systems are rules (or production rules), semantic nets, and frames [Waterman 86]. This system uses rules to state all the facts and relationships about the problem because they provide a natural way to describe processes and are easy to implement using the conventional program language, C.

Rules are represented in a format of

IF <conditions>
THEN <conclusions>

or in a format of

IF <conditions>
THEN <conclusions>
ELSE <conclusions>.

The conditions of the IF portion are performed or checked against the collection of knowledge. In the first format, if the conditions of the IF portion are satisfied then the conclusion are valid. In the second format, if the conditions of the rule are satisfied, the conclusions under THEN are valid. Otherwise, the conclusions under ELSE are valid. The conclusions made according to rules then become part of the collection of knowledge and may be used as conditions to "fire" other rules.

4.3 Rules for Features Recognition

This section will discuss how the rules are used to describe geometric shapes, topological relationships, feature recognition schemes, and search algorithms.

4.3.1 2D Drawings Data.

TABLE 2. SKETCH HISTORY

Order	Primitive	Number
1	line	1
2	line	2
3	line	3
4	line	4
5	line	5
6	line	6
7	line	7
8	line	8
9	line	9
10	arc	1
11	line	10
12	line	11
13	ellipse	1
14	circle	1

TABLE 3. DATA FOR 2D PRIMITIVES

LINE SEGMENT

No.	X1	Y1	X2	Y2	Pt1	Pt2
1	184	189	103	141	1	2
2	103	141	182	96	2	3
3	182	96	259	149	3	4
4	259	149	181	186	4	5
5	112	147	118	123	6	7
6	118	123	190	77	7	8

ELLIPSE

No	Cx	Cy	Maj	Min	Angle
1	186	208	21	12	0

ARC

No	Cx	Cy	X1	Y1	X2	Y2	r	Dir.
1	187	165	164	145	209	144	95	CCW

CIRCLE

No	Cx	Cy	r
1	185	221	12

The feature recognition system recognizes features based on the simple geometric primitives and topological information stored in the 2D drawings database which is created by the freehand sketching subsystem [Fang 88]. The freehand sketching system creates a database to maintain a sketch history and a database to record parameters of geometric entities such as line segments, ellipses, circles, and circular arcs. The sketch history records the sketching sequence. Table 2 shows the sketch history of Figure 4. First, the designer sketched nine line segments to form a block (1~9). Then he/she sketched an ellipse (13), two more line segments (11, 12), and an arc (10) to form a cylinder. Finally added a circle (14) on the top of the cylinder to represent a sphere. The database provides following the information on the features recognition subsystem: 1) the kind of entities which has been sketched, 2) how many of each kind of entity have been sketched, and 3) the order in which the entities have been sketched.

Table 3 shows another database: parameters of each primitive. The information in the database includes the serial numbers of the primitives and their parameters. The parameters of these primitives are similar to those of a traditional 2D CAD system and are used by the features recognition subsystem for the evaluation of the shapes and the dimensions of features. The parameters for the line segments are coordinates and the topology of two end points. The parameters for the ellipse are the center position, lengths of the major axis and the minor axis, and the angle of the major axis. The parameters for the circle are the center point and the radius. The parameters for the arc are the center position, the radius, the coordinates of two end points, and the sketching direction (clockwise or counterclockwise).

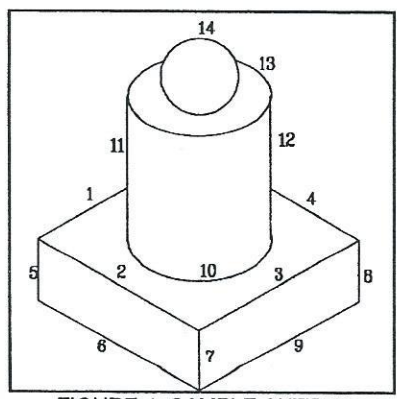

FIGURE 4. SAMPLE SKETCH

These data (Table 2 and Table 3) are used to determine geometric relationships among 2D primitives and are used to fire rules for feature recognition. To conclude the following rule, the system will search through data in the Table 2 to see if only *line* primitives exist.

IF

-the feature is composed of only line segments

THEN
-it is a line_structure.

To conclude the following arrow_head_structure (see Figure 8) rule, data in the Table 3 are used to determine if the two lines share an end point and are used to calculate line angles to determine if they are in principal directions and perpendicular to each other.

IF
-two lines share an endpoints AND
-the two lines are in principal directions AND
-the two lines are perpendicular to each other in the 3D coordinates
THEN
-it is a arrow_head_structure.

4.3.2 The Search Tree and Rules Derivation.

For the human visual cognition system, 3D features can be defined by 2D geometric entities such as vertices, edges (lines), and faces, with the proper topology. To a human being, a cylinder can be recognized by the presence of two ellipses and two line segments (see Figure 5b); This cognitive phenomenon is the basic concept for generating useful information (knowledge) for the recognition of features. For example, knowledge of recognizing three basic feature structures (spherical_structure, cylindrical_structure, and block_structure) could be delineated as follows:

1) A single circle defines a spherical_structure (Figure 5a).
2) Two ellipses, two line segments and the appropriate topology define a cylindrical_structure.
3) Nine line segments and the appropriate topology define a block_structure.

Based on this simple schematic, a tree to systematically search for features has been created (Figure 6). The search starts from the top of the tree then travels down to one of the seventeen features according to geometric conditions. The following sections describe the process required for the creation of the tree and the rules derived for features recognition.

4.3.2.1 Features Classification in Terms of Geometric Shapes.

All the features available in this system can be classified according to three basic feature structures: the block_structure, the cylindrical_structure, and the spherical_structure (Table 1). The wall and the rib are block_structures with specific thicknesses and heights. The boss and the hole are cylindrical_structures, but one is solid whereas the other is hollow. Similarly, other

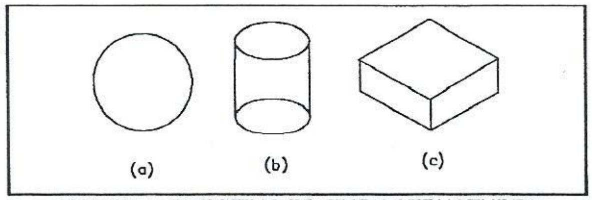

FIGURE 5. PRIMITIVE FEATURE STRUCTURES

features are classified according to the block_structure group, cylindrical_structure group, or spherical_structure group. Thus, the system commences by choosing one of the three groups for searching. A closer examination of those features found that features in the block_structure group are represented simply by line segments in human-understandable sketches. Features in the cylindrical group are represented by line segments and ellipses. Features in the spherical_structure group are represented by single circles. These facts are represented in the first three rules.

IF
-the feature is composed only of line segments
THEN
-it is a line_structure.

IF
-the feature is composed of line segment(s) and ellipse(s)
THEN
-it is a line_curve_structure.

IF
-the feature is composed of a single circle
THEN
-it is a circle_structure.

4.3.2.2 Shorthand Sketching.

Before travelling further into the search tree, the possibility of shorthand input of features must be discussed. Figure 7a shows that a complete block_structure is defined by twelve line segments and the proper topology. However, usually the three hidden line segments are not sketched since sketches with hidden lines removed are consistent with visual perception (Figure 7b).

In Current CAD systems, "extrusion" is a common way to create constant-cross-section 3D solids. The method is based on the notion of defining a closed profile, then pulling the profile along a path. Figures 7c-h illustrate process of constructing the block_structure by defining a rectangular cross section then pulling it along a principal axis.

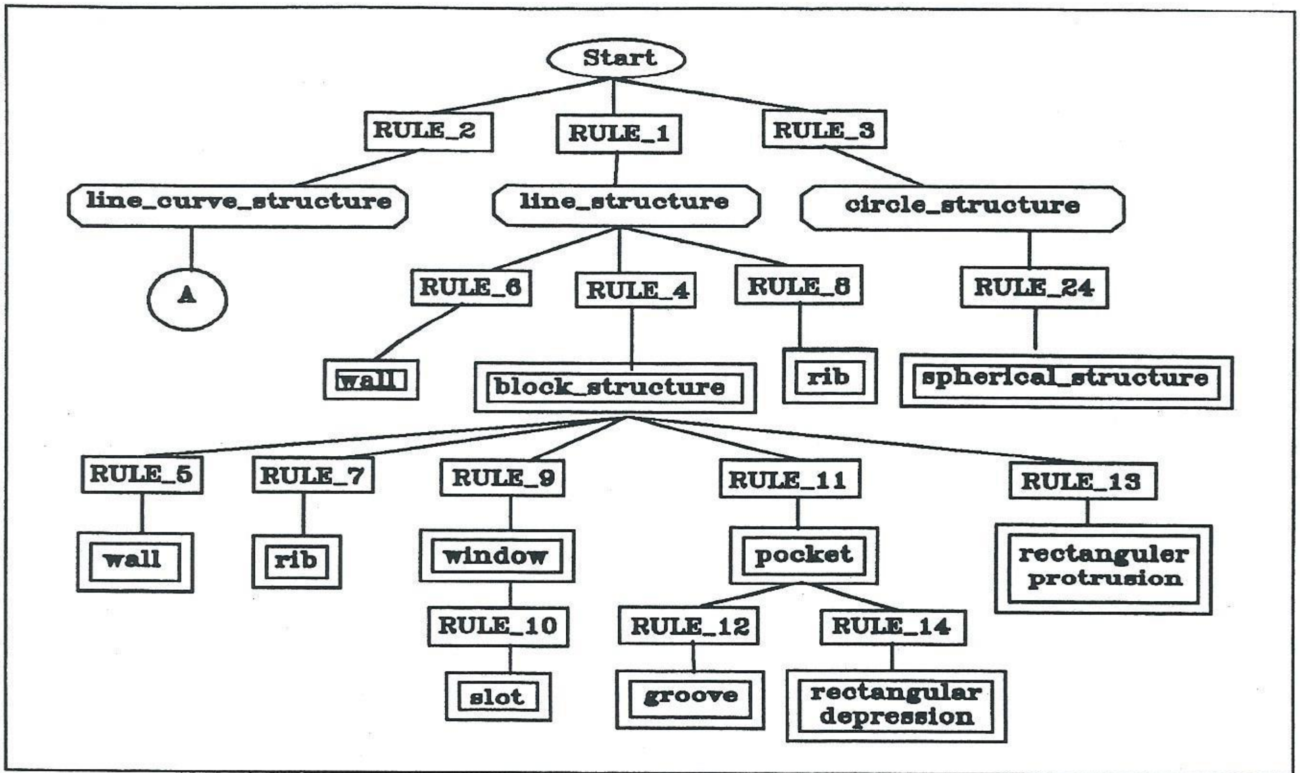

FIGURE 6. SCHEMATIC SEARCH TREE FOR FEATURES RECOGNITION

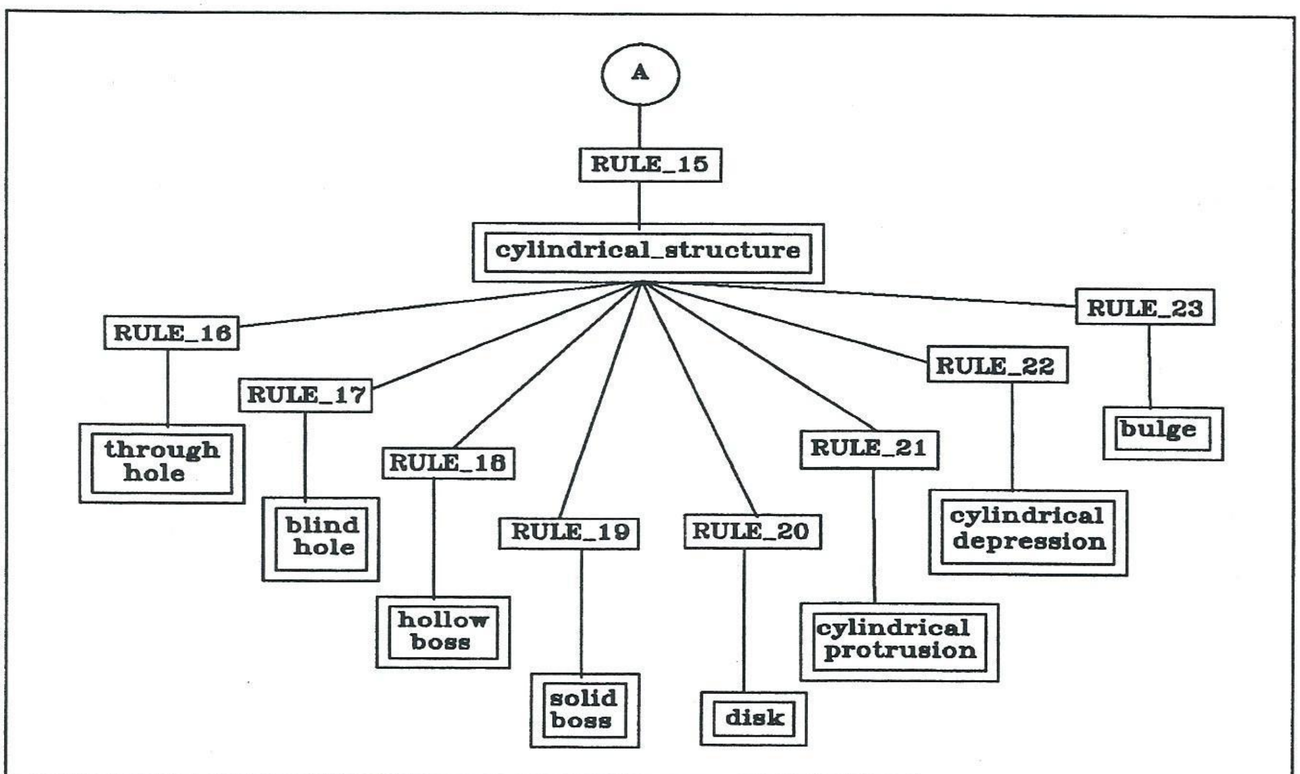

FIGURE 6. SCHEMATIC SEARCH TREE FOR FEATURES RECOGNITION (CONTINUED)

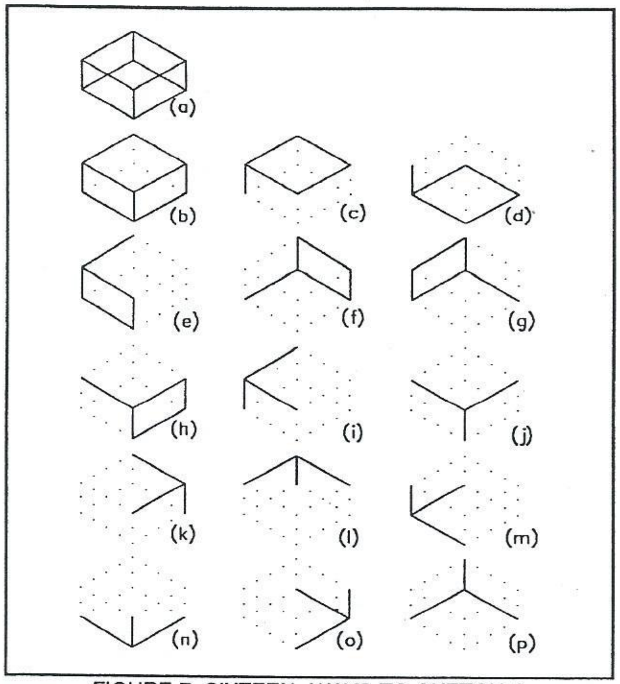

FIGURE 7. SIXTEEN WAYS TO SKETCH A BLOCK_STRUCTURE

From the geometric point of view, data required to define a block are length, width, height; therefore a block_structure can be defined as simply as three line segments which intersect at a point forming a vertex of the block_structure (Figure 7i-p). This kind of structure is called a "duck_claw_structure" in the rules. Similarly, the shorthand practice can be applied to other features.

4.3.2.3 Necessary and Sufficient Conditions for Defining Features.

Figure 7 shows sixteen ways to sketch the block_structure and these are not all the possible alternatives. This makes deriving rules for recognizing every possible sketch impractical. A more desirable strategy for recognizing features is catching those dimensions which are necessary and sufficient for defining them. For example, center position, length, width, and length are necessary and sufficient conditions for defining a block_structure in spatial coordinates. Therefore, an algorithm is designed to recognize the block_structure by searching through the 2D sketching database for length, width, and height of the block_structure then calculating the center of the block_structure. Once these data are obtained, other "redundant" information can be ignored. In other words, no matter how thorough the designer sketches a block_structure, the search scheme will just try to find a line segment for length, a line segment for width, and a line segment for height, then calculate the center position of the block_structure and fill in all other information. This strategy significantly simplifies the rule derivation.

4.3.2.4 The Block_structure Group.

All features in this group share a common geometric shape: block_structure. It is the basic geometric feature of this group. With different sizes and positions, "block_structures" play various roles in an injection molded part, therefore they have specific meanings to manufacturers and are given specific names, such as wall, rib, window, etc. Following is the example of recognizing a "wall".

If the search goes to the line_structure in the search tree(see Figure 6), the wall feature will be the first assumption since it is the most common feature for the domain of plastic injection molded parts. The second assumption will be the rib feature and the third assumption will be the slot feature, etc. There are eight features in the group. Rules to recognize them are discussed below. According to Figure 7, it seems that there are too many ways to define the block_structure. But notice that all processes contain at least one duck_claw_structure (Figure 8) and that the structure includes the necessary and sufficient conditions for defining a block_structure. Therefore, the rule to used recognize the block_structure is:

IF
-the feature is a line_structure AND
-the feature is a duck_claw_structure AND
-dimensions to define a block_structure have been found AND
-the spatial position of the feature has been found
THEN
-the feature is a block_structure.

The wall is virtually a block_structure with a thin thickness. All the processes used to define the block_structure can be used to define the wall. The characteristic used to distinguish the wall from the bock_structure is thickness. One of the two rules used to recognize the wall is:

IF
-the feature is a block_structure AND
-one of the three dimensions (length, width and height) of the block_structure is relatively small
THEN
-the feature is a wall.

According to the rules for designing injection molded parts, thicknesses of walls are usually constant so that a uniform material structure can be obtained. This simplifies the process of creating the wall by assigning the thickness of the wall a default value. Figure 9 shows the possible lines sketched that can be interpreted as an x-direction wall. Similar processes can be applied to create y-direction and z-direction walls. These wall sketches may not form a duck_claw_structures but are composed of at least one "arrow_structure" which is formed by two line segments with an acute angle. Another rule used to recognize the wall is:

IF
-the feature is a line_structure AND
-the feature is an arrow_head_structure AND
-dimensions to define a wall have been found AND
-the spatial position of the feature has been found
THEN
-the feature is a wall.

The other common features in the block_structure group are window and slot. Geometrically, the window is a rectangular through hole on the wall and the slot is a narrow window. Rules to recognize the window and the slot are:

IF
-the feature is a block_structure AND
-the third dimension on the feature is negative AND
-the third dimension is equal to or greater than the thickness of the parent feature
THEN
-the feature is a window.

IF
-the feature is a window AND
-the length/width ratio is greater than a certain number
THEN
-the feature is a slot.

4.3.2.5 The Cylindrical_structure Group

There are eight features in this group and they share a basic geometric shape: cylindrical_structure. Thus, the cylindrical_structure is the first goal to be recognized when search for features in the cylindrical_structure group. Figure 10 shows possible sketches for defining a cylindrical_structure. The sketches are all constructed with two entities, the ellipse and the line segment. The rule to recognize the cylindrical structure is:

IF
-the feature is a line_curve_structure AND
-the number of ellipses is less than 2 AND
-the number of line segments is less than 2 AND
-the line segment is perpendicular to the major axis of the ellipse AND
-dimensions to define the feature have been found AND
-the spatial position of the feature has been found
THEN
-the feature is a cylindrical_structure.

The hole and the boss are most popular features in the group of cylindrical_structure and are taken as examples of recognizing cylindrical features. Holes are cylindrical cavities in parent features. The process to sketch a hole is 1) sketching an ellipse on a face of the parent feature in order to define the diameter of the hole and 2) sketching a line segment down into the parent feature in order to define the depth of the feature. If the depth of the hole is greater than the thickness of the parent

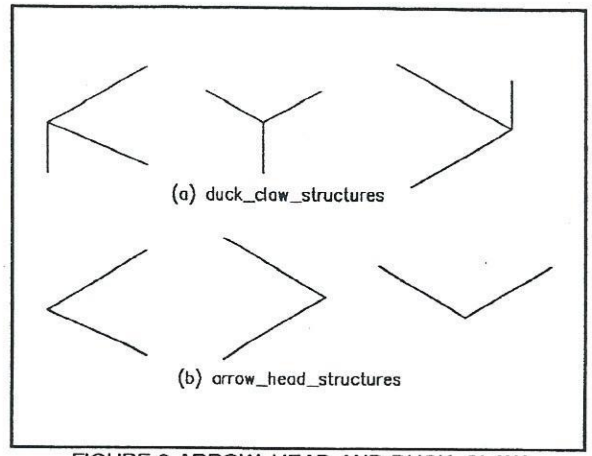

FIGURE 8 ARROW_HEAD AND DUCK_CLAW STRUCTURES

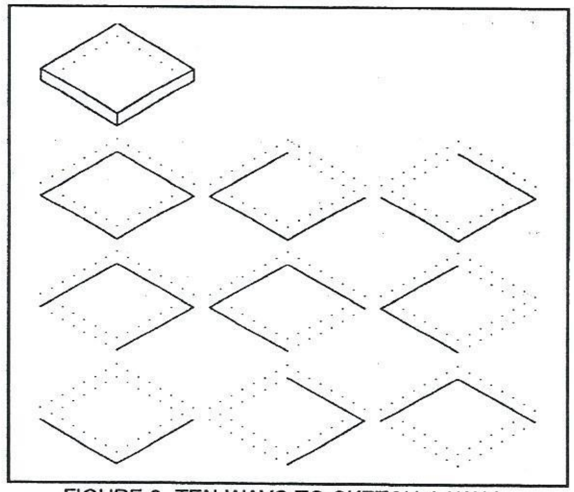
FIGURE 9. TEN WAYS TO SKETCH A WALL

feature, then it is a through hole. Otherwise, it is a blind hole. Rules used to recognize the through hole and the blind hole are:

IF
-the feature is a cylindrical_structure AND
-the length of the feature is negative AND
-the length of the feature is greater than the thickness of the parent feature
THEN
-the feature is a through hole.

IF
-the feature is a cylindrical_structure AND
-the length of the feature is negative AND
-the length of the feature is less than the thickness of the parent feature
THEN -the feature is a blind hole.

There are two kinds of bosses: the solid boss and the hollow boss. The hole in the hollow boss may be functionless but makes the wall thickness of the boss about equal to the thickness of the parent feature so the thickness is constant. The rule used to recognize the hollow boss and the solid boss are:

IF
-the feature is a cylindrical_structure AND
-the diameter of the feature is greater than three times the thickness of the parent feature AND
-the length of the feature is positive
THEN
-the feature is a hollow boss.

IF
-the feature is a cylindrical_structure AND
-the diameter of the feature is less than three times of the thickness of the parent feature AND
-the length of the feature is positive
THEN
-the feature is a solid boss.

4.3.2.6 The Spherical_structure Group

The spherical_structure is probably the most simple primitive to sketch and to recognize. A single circle defines the spherical_structure and the rule to recognize it is:

IF
-the feature is a circle_structure AND
-the diameter of the feature has been found AND
-the spatial position of the feature has been found

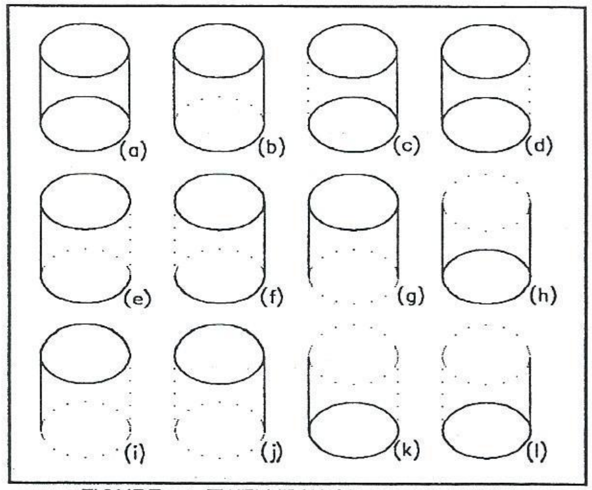

FIGURE 10. TWELVE WAYS TO SKETCH A CYLINDRICAL_STRUCTURE

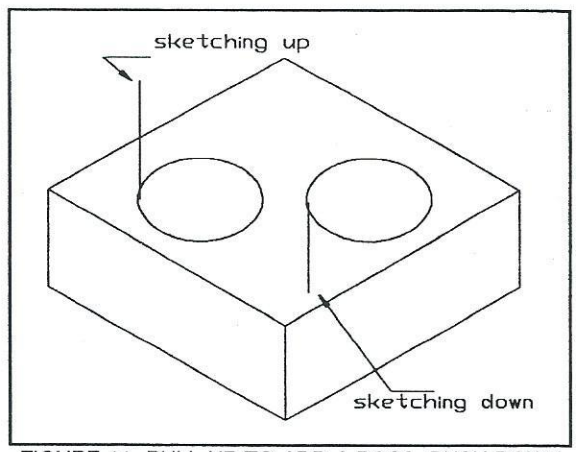

FIGURE 11. PULL UP TO ADD A BOSS. PUSH DOWN TO CUT A HOLE

THEN
-the feature is a spherical_structure.

4.3.2.7 Positive or Negative Volume

Positive or negative dimension determines whether the feature's volume is positive or negative. A positive volume block_structure includes features such a wall or a rib, and a negative volume block_structure includes features such as a window, a pocket, or a groove. Similarly, a positive volume cylindrical_structure includes features such as a boss or a disk, and a negative volume cylinder could include features such as a hole or a cylindrical depression. The very first feature is always interpreted a positive volume but the second and all subsequent features can be of either positive or negative volume. The rule to distinguish between them is the "direction of extrusion". To add a boss to an existing object, designers define a cross-section of the boss by sketching an ellipse on a surface of the object, then sketching the height of the boss outward from the surface in order to pull it up (Figure 11). To add a hole to the existing object, designers define a cross-section of the hole by sketching an ellipse on a surface of the object, then sketching the depth of the hole inward from the surface in order to make the hole.

IF
-the length of the feature has been pulled outward from the surface of the existing object
THEN
-the feature is of a positive volume.

IF
-the depth of the feature has been pushed inward from the surface of the existing object
THEN
-the feature is of a negative volume.

4.3.2.8 Approximation Rules for Sketch Inaccuracy.

When designers try to sketch a line segment which is parallel to one of the principal axes, the line segment, in all likelihood, will probably not be completely parallel to the axis due to the inaccuracy of sketching.

Approximation rules are, therefore, required for defining parallelism and perpendicularity. Figure 12 shows the isometric coordinates and the tolerance zones for line segments "parallel to" principal axes. The tolerances are plus and minus an adjustable parameter ($\in_1$). One of the approximation rules is as follows:

IF
-the line segment lies between $(90-\in_1)$ degree and $\{90+\in_1)$ degree
THEN
-the line segment is parallel to the y_axis.

Other than parallelism and perpendicularity, dimensions of new features are often explicitly or implicitly inherit from the parent feature. For example, the depth of the through holes, windows, and slots must be equal to the thickness of the parent feature; the length of walls and ribs are usually equal to the length (or width, or height) of the parent feature; the length of bulges are usually equal to the height of the parent feature. These "equal to" dimensions, in all likelihood, can't be sketched just equal to dimensions of their parent

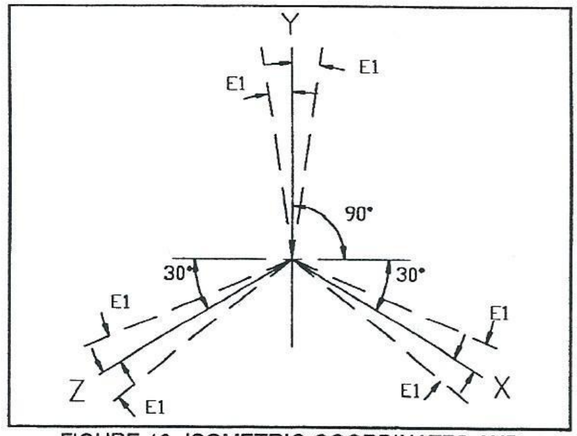

FIGURE 12. ISOMETRIC COORDINATES AND TOLERANCE ZONES

features, therefore, dimension rules are created as the example as follows:

IF
-the feature is a through hole OR
-the feature is a window OR
-the feature is a slot
THEN
-depth of the feature is equal to thickness of the parent feature

4.4 Inference Mechanism

The system doesn't have an isolated inference engine. The inference mechanism is incorporated with the knowledge in the program code. It supports forward-chaining and backward-chaining strategies for features recognition. For example, searching for a goal through the search tree in Figure 6, the inference mechanism assumes the block_structure, the cylindrical_structure, and the spherical_structure as intermediate goals and attempts to prove them by backward reasoning. The mechanism assumes the block_structure to be the first goal since the block_structure is the most common feature shape of injection molded parts. If the first attempt fails, the inference mechanism assumes a cylindrical_structure and tries prove the correctness of its second attempt. This process continues until a valid goal is reached or the possibilities are exhausted. Once an intermediate goal has been reached, the goal then will be used as a piece of data to drive continued forward searching for the final goal. Conversely, during forward-chaining, if any condition in the rule cannot be concluded from facts, then back-tracking will be evoked in order to obtain the conclusion of the condition.

4.5 Sample Result

Figure 13 shows a solid model of an injection molded part created with the Design Capture System. This part took the designer less than one minute to sketch. The processes are as follows:

- Sketching three lines which "perpendicular" to each other to create the thicker wall_1.
- Sketching two lines to create wall_2 with a default thickness.
- Sketching another two lines to create wall_3.
- Sketching an ellipse on wall_2 and sketching a line inward the wall to create hole_1.
- Sketching an ellipse on wall_1 and sketching a line outward from the wall to create boss_1.
- Create boss_2 with the same procedure.
- Sketching a line on wall_1 to create a rib with default thickness and height.

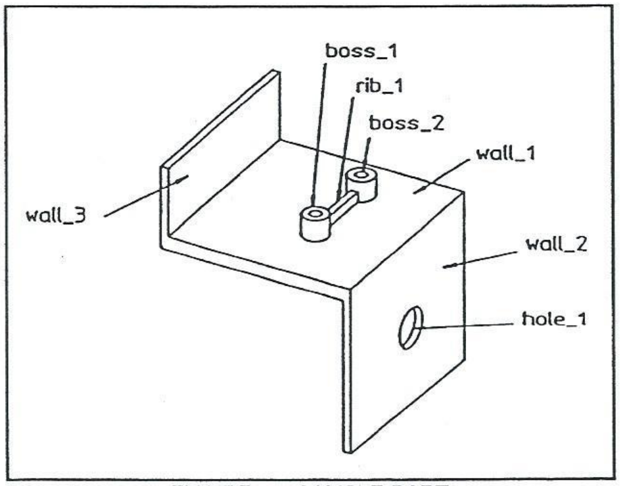

FIGURE 13. SAMPLE PART

5. CONCLUSIONS

The feature recognition system is one of the Design Capture system's three subsystems. It works with the other two subsystems, the freehand sketching subsystem and the spatial reasoning subsystem, to capture the designer's intent without burdening he/she with many unnecessary menu operations and intermediate steps as required by most CAD systems. It can be integrated with a more and more sophisticate variational geometry or parametric technique to build a conceptual design tool for mechanical design engineers.

The weakness of this system is that it recognizes finite number of features. To construct a complicated design, a large number of features are required. To solve this problem, the "design with feature" system should be able to provide sole a common feature set with a program which allows users to the particular set of features they need.

Another difficulty was found when performed system test. It is inconvenient to sketch on the tablet with a stylus while the sketched object is displayed on the screen. Designers felt inconsistent because they had to watch the object on the screen while performed sketching on "remote" tablet. A solution to this problem is sketching directly on the screen with a light pen.

ACKNOWLEDGEMENT

We gratefully acknowledge the financial support received from SDRC (Structural Dynamics Research Corporation) and from Schulumberge. We also acknowledge the equipment grant received from Hewlett Packard Laboratories.

REFERENCES

[Clowes 71] Clowes,M.B. On Seeing Things, *Artificial Intelligence*, Vol.2, pp.79-116, 1971.

[Fang 88] Fang,R.C. Ullman,D.G. Free-hand: A sketching Recognition System for Conceptual Mechanical Design, *Report, Design Process Research Group, Oregon State University*, 1988.

[Henderson 88] Henderson,M.R. Chang,G.J. FRAPP: Automatic Feature Recognition and Process Planning from Solid Model Data, *Computers in Engineering: Proceedings of ASME ICEC*, Vol.1, pp.529-536, 1988.

[Huffman 71] Huffman D.A. Impossible Objects as Nonsense Sentences, *Meltzer, B. and Michie,D. (eds) Machine Intelligence*, 6, pp.295-323, 1971

[Huffman 78] Huffman D.A. Surface Curvature and Applications of the Dual Representation, *Computer Vision System*, pp.213-222, 1978

[Hwang 90] Hwang, T.S. The Design Capture System: Capturing Back-Of-The-Envelope Sketches, *Doctoral Dissertation Submitted to Oregon State University*, June 1990

[Hwang 90a] Hwang,T.S. Ullman,D.G. The Design Capture System: Capturing Back-Of-The-Envelope Sketches, *Proceedings of International Conference on Engineering Design (ICED)*, Dubrovnik, Yugoslavia, Aug. 1990.

[Kanade 81] Kanade,T. Recovey of the Three-Dimensional Shape of an object from a Single View, *Artificial Intelligence*, Vol.17, pp.409-460, 1981.

[Lee 85] Lee,S.J. Haralick,R.M. Zhang,M.C. Understanding Objects with Curved Surfaces from a Single Perspective View of Boundaries, *Artificial Intelligence*, 26, pp.145-169, 1985

[Luby 86] Luby,S.C. Dixon,J.R. Simmons,M.K. Creating and Using Feature Data Base, *Computer in Mechanical Engineering*, Nov., pp.25-33, 1986.

[Pugh 89] Pugh,D. Viking: A Sketchpad for Creating Models of Three Dimensional Objects, *A Research Proposal for the PhD. Thesis, Carnegie-Mellon Univ.*, 1989.

[Sakurai 88] Sakurai,H. Gossard,D.C. Shape Feature Recognition: A Bridge between Symbolic and Numeric Computing Environment, *Intelligent CAD: Proceedings of The Second IFIP WG5.2 Workshop*, pp.425-433, 1988.

[Sanker 77] Sanker,P.V. A Vertex Coding Scheme for Interpreting Ambiguous Trihedral Solids, *Computer Graphics and Image Processing*, Vol.6, pp.61-89, 1977.

[Shah 88] Shah,J.J. Wilson,P.R. Analysis of Knowledge Abstraction, Representation and Interaction Requirements for Computer Aided Design, *Computers in Engineering: Proceedings of ASME ICEC*, Vol.1, pp.17-25, 1988.

[Shah 89] Shah,J.J. Philosophicl Development of Form Feature Concept, *Proceedigns of NSF Engineering Design Research Conference*, pp.317-332, 1989.

[Shapira 85] Shapira,R. More about Polyhedral: Interpretation through Constructions in the Image Plane, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.1, PAMI-7, pp.1-16, 1985.

[Sugihara 78] Sugihara, K. Picture Language for Skeketal Polyhedral, *Computer Graphics and Image Processing*, Vol.8, pp.382-405, 1978

[Sugihara 84] Sugihara,K. An Algebraic Approach to Shape-form-image Problems, Artificial Intelligence, Vol.23, pp.59-95, 1984.

[Sugihara 86] Sugihara,K. *Machine Interpretation of Line Drawings*, Cambridge, Masssachusetts: The MIT Press, 1986.

[Ullman 87] Ullman,D.G. Dietterich,T.G. *Mechanical Design Assistant Development: Program Background and Proposed Research*, Design Process Research Group, Oregon State University, Corvallis, Oregon, 1987.(UnPub)

[Vaghul 85] Vaguhl,M. Dixon,J.R. Zinsmester,G.E. Simmons,M.K. Expert System in CAD Environment: Injected Molding Parts as an Example, *Computers in Engineering: Proceedings of ASME ICEC*, pp.72-78, 1985.

[Waterman 86] and Waterman, D.A. Chapter 7. In: *A Guide to Expert System*, Reading, Massachusetts: Addison-Wesley Publishing Co., 1986, p. 63-79.

[Wentorf 89] Wentorf,R. Shephard,M.S. Korngold,E.V. The Use of Functional Models to Control Engineering Idelizations, *Computers in Engineering: Proceedings of ASME ICEC*, 1, pp.63-70, 1989.