Detection and Evaluation of Orientation Features for CAD Part Models

M. I. Kilani, R.H. Sturges

EDRC 24-66-91

Detection and Evaluation of Orientation Features for CAD Part Models

Mohammad I. Kilanl
Graduate Research Assistant

Robert H. Sturges
Assistant Professor

Department of Mechanical Engineering
and Engineering Design Research Center of
Carnegie Mellon University
Schenley Park, Pittsburgh, PA 15213-3890
412-268-3657

Abstract

Current design for assembly methodologies stress the importance of part features related to acquisition and orientation in determining time values and error rates. This paper discusses an approach to detecting and evaluating features of CAD modelled parts which contribute to assembly difficulty. An evaluation basis is described and algorithms are developed which return an index of difficulty with respect to orientation features and degree of symmetry for two- and three-dimensional parts. Limitations of the method are discussed with examples.

1. Introduction

Design for assembly (DfA) methods have grown steadily in capability and application popularity for at least the last decade as designers and manufacturers work to rationalize the assembly process, usually by redesigning their products tc facilitate assembly [Miller, 1988], [Rooks 1987). These techniques are either primarily qualitative [Miles 1982] or quantitative [Boothroyd 1987, 1988]. Using these methods, several design improvements may be obtained, including reducing the number of parts required, shortening the assembly time and reducing the complexity of the assembly machines [Schuch 1989]. These methods have been usefully applied in various industries with product line ranging from electrical products [Funk 1989] to aerospace structures [Baum 1989]-

There now appears a strong demand for integrating DfA knowledge into computer-aided design systems [Libardi 1988]. Given a design of a Mechanical System Assembly (MSA), the objective is to develop an "assembly critic" which will evaluate the design and will recommend improvements based on the evaluation results. From such an integrated system, the designer would receive feedback on the degree of difficulty associated with the proposed design and recommendations on design modifications that would improve assemblability while maintaining the original function.

This paper will briefly review models of assembly difficulty and the related concerns of component, process, and system analysis of assembly. We observe from such models that a significant portion of total assembly effort derives from orientation feature subtlety and the degrees of symmetry possessed by a part. Methods for evaluation of orientation feature subtlety will be presented in section 3. Shape extraction with symmetry evaluation will be presented in section 4 with examples. Finally, we will discuss the reasoning base needed for evaluating parts which combine orientation features and symmetry in section 5.

2.0 Models of Assembly Difficulty

The degree of difficulty associated with assembling a mechanical system has been the subject of several research studies in assembly evaluation and automation. Boothroyd and Dewhurst [Boothroyd 1983] developed a quantitative, empirical, manual method for evaluation which utilized assembly charts. Another method for assemblability evaluation has been developed by Hitachi [Miyawaka 1986]. In both of these methods, an assembly evaluation form is completed by entering the part names and numbers in the order of assembly. Scores are determined for each part and the evaluation score is correlated to an assembly cost ratio. In other research [Sturges & Wright 1989], a predictive analytical model based on human motor capacity quantifies an *index of difficulty* for most of the factors affecting assemblability. This index of difficulty (ID) measures the dexterity and the time required to assemble the part.

Another area of research that is closely related to assemblability evaluation is the representation and determination of part features from a computer model. An attempt to extract *form features* (volumes removed from the stock of material by machining operations) from a boundary representation (B-rep) model is presented in [Henderson 19841. The approach is to search the product model for some relevant patterns which define features. This approach, however, was found to be computationally expensive. To avoid the difficulties associated with feature extraction, Dixon [Dixon 1988] suggested a feature based representation scheme which represent parts in terms of their features.

According to the ID and other models, assembly tasks are divided into gross motions and fine motions. We adopt the terms acquisition phase and assembly phase, respectively. A study of a number of example products has shown that ten significant measurable factors account for almost all the task time in the assembly [Sturges, et al.,1986]. Of these, the dominant factors affecting difficulty during acquisition are orientational feature subtlety and part shape. These two factors account for about half the time spent in the bench assembly acquisition phase. Significantly, these have the opportunity of being reduced to near zero by pre-orienting the parts, a technique widely employed in electronics manufacturing, but seldom applied to typical mechanical assemblies.

In describing the factors affecting assemblability, we differentiate between *component-level* factors, *system-level* factors

and *process-level* factors. Component-level factors pertain to the **individual components** of the assembled product and can be identified by **examining the** components independently of the overall system. System-level factors depend on the interactions between the components. Process-level factors depend on the process employed to assemble the product. Table 1 shows thesg assemblability factors organized with respect to associated phase and level. We differentiate the various levels to identify the models required during the design stage. Also, the assembly difficulty is usually based on an ideal geometric relation between the task and the effector. Substantial increases in difficulty can be caused by working conditions which are less than ideal [Sturges 1990]. A model of the assembly process which includes the task/effector relationship is required to account for this increase.

The effects of system-level and process-level factors on assembly difficulty are greatly influenced by the assembly sequence selected. The generation of these sequences are necessary input to orientation analysis and are beyond the scope of this research. The reader is referred to [De Fazio, 1987] for a discussion of mechanical assembly sequences.ed.

For the purposes of this research, the inputs to an orientation feature detector and evaluator consist of solid or planar representations of parts and a knowledge base modelling the effector characteristics. The task is presumed to be that of orienting and aligning the part with respect to a sub assembly. Other factors such as handling distance and grasping are considered in [Sturges 1989] and not included here. The outputs of the detector/evaluator consist of an ID and **a** set of recommendations for part design improvement with respect to assembly.

The additional concern of assembly errors due to orientation features has not been modelled. Examples from field experience may serve to build a knowledge base of "lessons learned" to avoid hidden cost drivers. For example, a major domestic motor manufacturer employed an oil wick design which was asymmetric (Figure 1.) The wick could be assembled into the motor in either orientation, but functioned properly only with the tip down. Field failures of the motors began to appear many months after the design was introduced, and thousands of units were already in use by customers.

3. Difficulty Due to Orientation Features

The purpose of feature subtlety extraction is to measure the difficulty in determining the orientation of a part, e.g., on which end of the bolt is the head. In general, we evaluate the difficulty in finding those features of a part which orient it. This difficulty is in contrast to the amount of rotation needed to bring the part into alignment, which will be discussed in the next section.

We need to determine two facts when orienting a part:

1. **How** the top is distinguished from the bottom, or, how much one **end** differs from the other end.

2. How the part is oriented about an axis.

The significant external features of a part which contribute to orientation difficulty are its bosses and grooves. Printed or painted markings and holes are not considered since they do not fall into the difficulty model. The index of difficulty, in bits, is found from the boss or groove size, w, and the overall size of the part in the same direction, from:

$$ID = \log 2(s/w) \qquad (1)$$

For a part with more than one feature, the value which presents the *most* difficulty is initially reported. The designer may chose to ignore this value for functional reasons, in which case the next most significant value is reported, etc. For example, a BIC™ pen can be oriented end to end by observing the difference in diameter between the cap and the body of the pen. If the cap is 10 mm in diameter and the body is 8 mm, then the boss, w, will be half the difference, or 1 mm. The ID for is found to be 2.33 bits. The assembly time is related to the ID by a constant factor dependent on the effector in use. The pen may be oriented for "assembly" to your pocket by observing the clip molded onto one side of the cap. It protrudes about 3 mm from the body of the pen which is about 13 mm across at that point. The boss value, w, is now 3, while the overall size, s, will be 13. The ID for axis **orientation is** 2.12 bits. Since this result is less than that for end to end orientation, the latter larger value is reported.

Two approaches are employed to extract orientation features from an object model: recursive subdivision, and pattern matching.

3.1 Recursive Subdivision

In this approach, bosses and grooves are found by recursively subtracting the part model from its bounding box, i.e., the smallest rectangular prism which contains the part. It is assumed that the part model is pre-oriented to the same axes as the bounding box. The actual part would be assumed to lie in a random orientation. First, an attempt is made to subtract the part from its bounding box. Resulting solid elements which do not lie on the surface of the box are presumed to be internal features, e.g., holes, and are discarded. If an empty set results, the part is known to be a simple rectangular prism with no bosses or grooves. If the result is a nonempty set, the remaining set of elements is subtracted from its own bounding box. The distance between the first and the second bounding box is considered a relative boss. If the results of this second subtraction is an empty set, then the elements resulting from the first are considered to be the grooves of the original model and no further subdivision is performed, as in Figure 3.

If the result of the second subtraction is a nonempty set, this second entity is a candidate boss. To verify this, the resulting elements are **subtracted from their** own bounding box. If an empty set results, the elements of this second subtraction are taken to be bosses. Those bosses not on the original bounding box surface are ignored. The subtraction process continues recursively until an empty set is obtained. Non-rectangular elements produce successive bounding boxes that equal to each other, and the process terminates on a curved or angled boss or groove element. Any element that gives an empty set for an odd numbered subtraction is considered a candidate groove; even numbered subtractions indicate bosses.

After all bosses and grooves are found, a second round of subdivision is made in which all these previously found elements are first subtracted from the original object. This second round identifies grooves such as shown in Figure 4. The resulting ID for an object is a set computed from equation (1) above for each boss and groove found. The maximum value and the element responsible is returned.

3.2 Pattern Matching

The extraction of "form features" from a boundary representation (B-rep) models is presented in [Henderson 1984). A form feature is defined as a volume or volumes of material removed from the stock material (e.g., by machining operations). This technique may be employed to find the parts smallest and largest dimensions in order to determine the effect of part size and to find the bosses and the grooves in the part and their dimensions. However, a system model is necessary to determine which boss or groove will affect the difficulty level.

In feature-based design systems, feature interaction is still a difficult problem. That is, even if a system is capable of recognizing two adjacent grooves, it may not be able to recognize a wall or a boss that resulted from the adjacency property of the grooves, shown in Figure 3. Some of the work being done in graph-based topological grammars may solve this problem in theory, but practical solutions are not close at hand [Finger and Dixon 1989]. We have adopted a variation of boundary representations that portrays non-manifold objects (illegal solids with dangling lines or surfaces) [Gursoz 1989] to facilitate feature extraction, because extra information about the solid can be included (e.g., center lines or cutting planes).

Our pattern matching approach finds bosses and grooves by matching a set of surfaces and edges in the model to a predefined feature. We call this approach incremental establishment of abstract concepts (IEAC) since it starts with less abstract elements, such as lines and surfaces, and incrementally establishes more abstract ones, such as bosses and grooves. The definition of an abstract feature specifies a flexible set of geometric and topological relationships to be satisfied by a corresponding set of elements in the part model. Figure 5 shows the relationship which defines a flat-bottomed groove. This feature specifies one surface of the part at a relatively lower elevation that the surrounding surfaces, and that there exists two other surfaces which connect the lower surface to the upper ones. Similar

relationships are used to define a boss. A production language tool (OPS5) **was used** to implement this method. Once definitions of the bosses **and** grooves **are** established and **a model part** is introduced, the inference engine attempts to match the objects of the model with the feature definitions. If a match is found, a production rule labels the feature. The dimensions of all labelled features are then applied to the ID equation (1) and the most significant value and feature are returned.

The rule-based orientation feature extractor identifies four different features: shoulders (Figure 6). grooves (Figure 7), bosses (Figure 8) and open holes (Figure 9). Blind holes and other features are beyond the scope of the present method. Surfaces, lines and relations between them are part of the input model, relieving the feature extractor from the surface recognition task. The relations between lines and surfaces are represented by a defined class of elements made up of incident-lines and associated-surfaces. This object class allows one to create arbitrary associations between lines and surfaces to represent solids with a variety of features. Additional feature classes were created for shoulders, bosses, grooves and holes. At the present time, we are limited to rectangular surfaces which are orthogonal to each other and parallel to the major x, y and z axes.

The algorithm execution is completely opportunistic state driven with no additional explicit control strategy. Initially, a bounding envelope for the part is established, as in recursive subdivision. Surfaces matching the definition of shoulders are then identified.

Shoulder Information is then used to detect bosses and grooves. Collections of bosses and grooves with certain relationships are then recognized as open holes. An exception-driven technique is used to refine the results of the feature extraction rules, since the same feature may be discovered more than once. Redundant features are removed by other rules. Naming rules for working memory elements were implemented externally in common LISP. An outline of program operation for each object class are given in the Appendix.

4. Difficulty Due to Part Shape

Part shape adds difficulty proportionate to the time needed to rotate a part about its several axes in order to align it for assembly. In bench assembly, two or more seconds of additional handling time are needed per part if it possesses no symmetry. If the part has been pre-oriented such that no further rotation is needed once it has been grasped, then no difficulty is added.

For example, the object with the greatest degree of symmetry is spherical: no preferred orientation, no difficulty in orientation, not generally useful. The object class with the least restrictions on orientation which embodies useful parts is the cylinder: no end-to-end preference; no axial orientation preference. A plain shaft, or even one with stepped ends if the steps are the same on both ends, is in this class. A washer is just a flattened cylinder; the aspect ratio (length/ diameter) has little or no effect on the difficulty in orienting this kind of part.

We will consider here the symmetry of planar figures. Solid objects of the same or lower symmetry are obtained by extrusion, revolution about an axis, or construction as a b-rep from such figures. Practical tests are not yet developed. When we say that a figure is "symmetrical" we mean that we can apply certain rigid transformations, which leave the whole figure unchanged. Planar figures possess two distinct symmetrical transformations: rotation about a point in the plane of the figure and reflection around an axis in the plane of the figure. Reflection around an axis is analogous to flipping the figure around that axis. Figures lO.a - lO.c show these symmetry transformations applied to a planar figure. In each of these transformations, the figure remains the same while the vertices switch places.

The evaluation of the symmetry of an object can be a very expensive computational process. To avoid the computational expense associated with symmetry evaluation, a set of rules have been developed which helps to narrow down the search space significantly. These rules are translation of geometrical facts about axes of symmetry and their relationships to each other and to other local properties of planar figures [Coexter 1969).

4.1 *Sytoxttttiy* Evaluation Rules

The following rules apply for planar figures represented using a linear B-rep scheme.

- The center of rotational symmetry is the center of area of the planar figure.
- The minimum possible angle of rotational symmetry is equal to the angle subtended between the center of area and two adjacent points intersecting a circle whose center is the center of area.
- The number of vertices to the left of an axis of symmetry is equal to the number of vertices to the right of that axis of symmetry
- The maximum number of axes of symmetry in a planar figure is equal to the number of edges in the figure.
- An axis of symmetry passes through a vertex or it bisects an edge.
- Axes of symmetry passing through a vertex bisect the angle at that vertex
- Axes of symmetry bisecting an edge are perpendicular to that edge.

In addition to the above relationships, group theory [Rosen 1983] states that the symmetry operations of any figures form a group called the *symmetry group* of the figure. Using group theory, the following useful relationships between the axes of symmetry are identified:

- All **axes** of symmetry intersect at a single point, which is the center of area for the planar figure.
- The angles between the any two adjacent axes of symmetry are equal. Also, the angle between two adjacent axes of symmetry, 8, is given by:

$$0 = \frac{180}{n},$$ where n is the number of axes of symmetry.

- A figure possessing axial symmetry possesses rotational symmetry with a minimum angle equal to double the angle between the axes of symmetry.

4.2 Symmetry Evaluation Algorithm

A symmetry quantification algorithm for 2-D planar figures is described below, using an enumerated, ordered vertices technique. The algorithm makes the following assumptions:

1. Planar figure (Figure 11, for example)
2. Straight line segments
3. Enumerated, ordered vertices within the figure
4. No internal features

Setup:

Given a set, M, of ordered indexes 0. . . . . .m-1 defining a planar figure with m vertices.

1. **Establish the set of potential symmetry axes nodes, N, where N contains all the points in M plus the mid-points between each two points in M. The indices of N are**

   0, 1, 2. . . . .2m-l = 0. 1, 2, ... n. where n = 2m-l

   N i = M i/2 for even i

   N i = Mid-point (M (i-l)/2 + M(i+l)/2) for even i.

2. **Establish potential symmetry axes starting points: the set of points with indexes 0, 1, 2, ... m-1.**

3. **Establish potential symmetry axis end points:**
   For each axis starting at point i, its end point index will be i+m.

Axes *Testing Procedure:*

Given a potential axis of symmetry, with

starting point index: i

end-point index: i+m.

test the points with indexes: i+1. i+2. . . . . .i+m-1 (group 1). against **the points: Mod[(l+n) / (n+l)l (group 2):**

i+1,	**against the points**	**Mod((i+n) / (n+1))**
i+2.		**Mod((i+n-1) / (n+1)**)
.		.
.		.
i+m-1		Mod(**(i+n-m+2)** / (n+1)).

Final Testing Procedure:

1. Shift **all** the points so that the starting point of the axis to be tested passes through the origin

2. Determine the reflection matrix for the axis

	f 2cos29 - 1	2cos9 sinG	1
H -	**I**		**I**
	L 2cos8 sin9	2sin29 - 1	J

where 0 is the slope angle of the axis being tested.

3. Multiply each point in group 1 above by the reflection matrix and compare with the corresponding point with the index given in group 2. Apply a tolerance policy depending on implementation.

Examples of the algorithm are given in Figure 12. The ID is computed proportional to the resulting class of symmetry obtained.

5. Combined Feature-Symmetry Difficulty

The foregoing methods have been shown to be effective in detecting and evaluating a class of single features which present difficulty in orientation during the acquisition phase of assembly. Since the basis for difficulty is measured in units of information, the measure is applicable to a range of effectors for which a relationship between ID and time can be established. The feature/part class is restricted to objects with little or no curvature, and in which internal features such as holes and painted markings are ignorable.

The foregoing methods are not applicable, however, to another class of features which are important in practice. This type of orientation feature is represented by the interaction between bosses/grooves and the symmetry of the part. Perhaps the simplest example of combined feature-symmetry difficulty is the part in Figure 13: a nearly square object. There are no external features, and the part is *nearly* 4-way symmetric. The ID based on orientation difficulty for this case is found to be:

$$ID = \log_2 (b / b - a) \qquad b > a . \qquad (2)$$

As **the** difference between the length and width gets smaller, the ID **increases** to a limit based on the tolerance policy. Beyond that, the ID drops to zero, as the length and width are essentially equal. This simple case is presently handled by an *ad hoc* rule, viz.: an aspect ratio test, but it fails for the next level of complexity which typifies the combined feature-symmetry class. Such parts are shown in Figure 14. In these cases, the difference between external features combined with the near-symmetry of the outline create orientation difficulty. The external feature size (boss or groove) is found by subtracting the part from its mirror image based on the symmetry axes of its bounding box. The resulting collection of objects form the difference set (Figure 15), each element of which is evaluated by equation (1) after elimination of its redundant twin.

6,0 Discussion

The detection and evaluation methods presented here are able to isolate a given feature within a certain class of features and to report the difficulty to a designer. This work has enabled us to automatically analyze, in part, two of the ten major assembly factors identified by the ID model. Recommendations are currently based on norms established for manual assembly. For example, features with total acquisition difficulty leading to a task time greater than 3 seconds are reported. The greatest single factor, say symmetry, is highlighted with a message such as, "Please try to reduce the asymmetry of this part," and the difference set is displayed if it exists. Where asymmetry is essential to function, such a blind recommendation makes little sense.

The generation of intelligent suggestions requires a larger knowledge base than component-system-process information which is isolated from *part Junction.* The representation of design intent in the form of a knowledge base which includes part function is clearly needed.

7. Appendix

7.1 Shoulders

As shown in Figure 6, a shoulder is a relationship among two surfaces and a line. The rules used to find a shoulder attempt to match working memory elements (WMEs) that satisfy this relationship. Once an instance is found, a new WME of the shoulder class is created. In addition, new WMEs to represent the association between the newly created shoulder and the surfaces participating in the instantiation are also created.

7.2 Grooves

As shown in Figure 7f a groove is a relationship among two shoulders and a common surface. Once an instance satisfying these conditions is found, a new WME of the groove class is created. In addition, new WMEs to represent the association between the newly created groove and the shoulders participating in the instantiation are also created.

7.3 Bosses

As shown in Figure 8, a boss is a relationship among two shoulders and a common surface. Once an instance satisfying these conditions is found, a new WME of the boss class is created. In addition, new WMEs to represent the association between the newly created boss and the shoulders participating in the instantiation are also created.

7.4 Holes

As shown in Figure 9, a hole is a relationship among two grooves and a pair of common surfaces. Once an instance satisfying these conditions is found, a new WME of the hole class is created. In addition, new WMEs to represent the association between the newly created holee and the grooves participating in the instantiation are also created.

8. References

Andreason, M.M., Kahler, S. and Lund, T. (1988), *DesignJetAssembly,* 2nd ed..Springer Verlag, New York.

Baum, J. P. and Gabitele, G. A. (1989), Design for Assembly of aerospace structures. A qualitative interactive approach," SME Technical Paper (series) MS89-158.

Boothroyd, G. and Dewhurst, P. (1983), *Design/or Assembly* -A *Designer's Handbook,* Dept. of Mech. Eng., Univ. of Massachussets, Amherst, MA.

Boothroyd, G. and Dewhurst, P. (1987), *Product Design for Assembly,* Boothroyd Dewhurst Inc., Wakefleld. RI.

Boothroyd, G. (1988), 'Making it Simple: Design for Assembly," Mechanical Engineering, Vol. 110, No. 2. pp 28-31.

Coexter, H.S. (1969), *Introduction to Geometry,* John Wiley & Sons Inc. 2nd Edition.

De Fazio, T.L. and Whitney. D.E. (1987) "Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation,* Vol. RA-3, No. 6, pp. 640-658.

Dixon, J.R (1988), "Designing with Features: Building Manufacturing Knowledge into More Intelligent CAD Systems," *Proceedings of ASME Manufacturing-International 88,* Atlanta. GA, April 17-20.

Funk, J. L., (1989), "Design for Assembly of Electrical Products," *Manufacturing Review,* Vol. 2, No. l,pp 53-59.

Henderson, M.R and Anderson, D.C. (1984), "Computer Recognition and Extraction of Form Features: A CAD/CAM Link," *Computers in Industry,* Vol. 5, pp. 329-339.

Khosla, P.K. and Mattikall. R, (1989), "Determining the Assembly Sequence from a 3-D Model," *Journal of Mechanical Working Technology,* Elsevier Press, June.

Ubardi, E. C, Dixon, J. R and Simmons, M. K. (1988), "Computer Environments for the Design of Mechanical Assemblies: A Research Review," *Engineering with Computers,* No.3,pp 121-136.

Miles, L.D. (1982), *Techniques of Value Analysis and Engineering,* 2nd ed. McGraw-Hill.

Miller. Fredrick W. (1988) "Design for Assembly: Ford's Better Idea to Improve Products," Manufacturing Systems Vol. 6, No. 3, pp 22-24.

Miyawaka, S. and Toshirjo, O. (1986), "The Hitachi Assemblability Evaluation Method (AEM)," *International Conference on Product Design for Assembly,* April 15-17, Newport, RI.

Myers, W. L.. Dixon J. R and Simmons, M.K. (1987), "Computer Analysis of Mechanical Assemblies from a CAD Data Base: Mechanical Handling Times," *Proceeding of ASME Computers in Engineering Conference,* NY, Aug. 9-13.

Rooks, B. (1987), "Danish Innovation Focuses on Design for Assembly," Assembly Automation, Vol. 7, No. 1. Feb.. pp 11-13.

Rosen, J., (1983), *A Symmetry Primerfor Scientist* John Wiley & Sons Inc.

Schuch, L. K. (1989). ftDFA Promises and Delivers;[1] *Assembly Engineering.* Vol. 32, No. 5, **May. pp. 18*21,**

Sturges, R.H., Dorman, J.G., and Brecker, J.N. (1986), 'Design for Producibility/[1] Westinghouse Productivity and Quality Center, P O Box 160, Pittsburgh, PA 15230-0160.

Stuiges, R E , and Wright, P.K. (1989), "A Quantification of Dexterity," *Journal of Robotics and Computer Integrated Manufacturing.* **Vol. 6, No. 1, pp 3-14.**

Sturges, RH. (1989), "A Quantification of Manual Dexterity. The Design for Assembly **Calculator,'*** ***Journal of Robotics and Computer Integrated Manufacturing,*** **Vol. 6, No. 3,** pp. 237-252.

Sturges, RH. (1990), "A Quantification of Machine Dexterity Applied to an Assembly **Task,"** ***International Journal of Robotic Research*** **Vol. 9, No. 3, pp. 49-62.**

Table 1. AstemblabUity difficulty factors, organized with respect to associated phase and level

	Acquisition phase: Parts brought from the feeding point to the assembly point.	**Assembly phase:** Parts are joined.
Component Level Factors: Factors pertaining to individual component independent of the overall system.	• Part Size: Difficulty in handling small or thin parts. • Part Shape: Difficulty in rotating the part about different axes to align for assembly.	• Stability: Difficulty represented by parts which must be restrained or require extra manipulation (e.g., parts that need to be held down temporarily or flexible parts).
System Level Factors* Factors dependent on the interaction between components.	• Boss or Groove Size / Feature Size: Difficulty in determining the correct orientation for the part to be acquired.	• Clearance: Difficulty that arises from the relative clearance between mating parts. • Direction: Difficulty arising from having to move the part in various insertion directions during assembly.
Process Level Factors: Factors dependent on the process employed to assemble the mechanical system.	• Handling Distance: Difficulty associated with bringing the part from the feeding point to the assembly point. • Handling Conditions: Difficulty due to weight, environment, tools, nesting and tangling.	• Fastening Method: Difficulty presented by the method used to fix the part for assembly. • Assembly Path: Difficulty associated with constrained paths the component follows during assembly process.

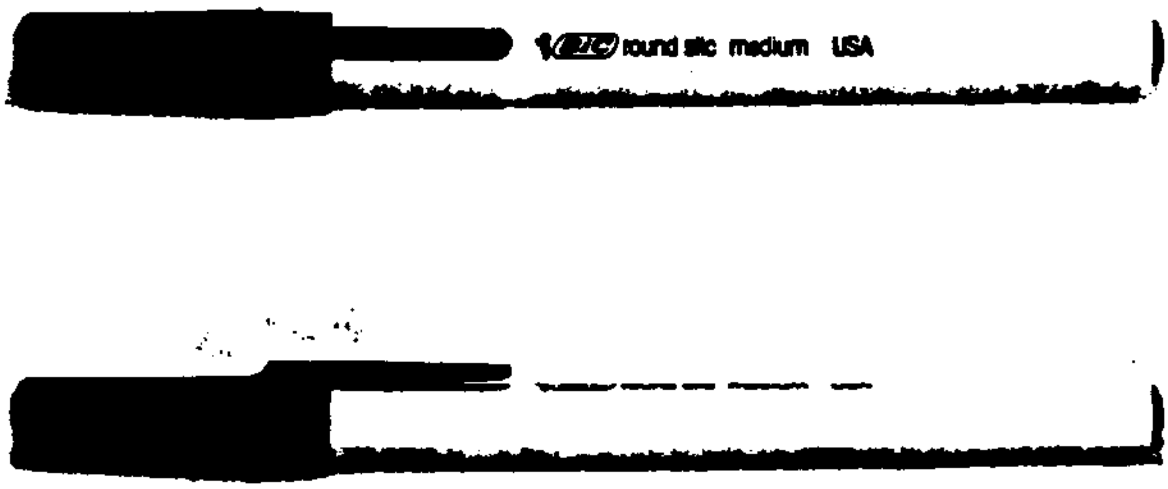


Figure 2. Two views of a BIC™ pen showing orientation difficulty features

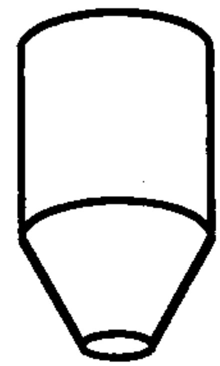


Figure 1. Asymmetric oil wick design

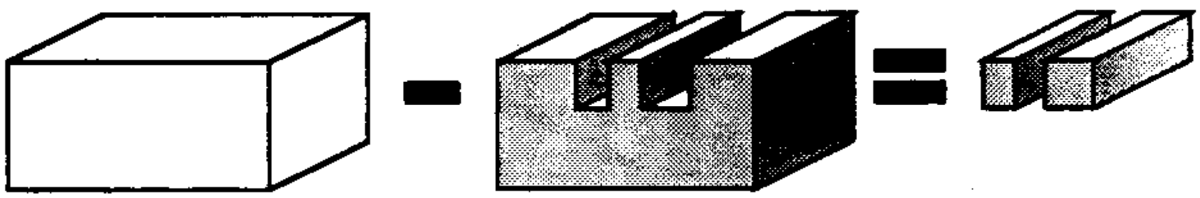


Figure 3. Part grooves obtained in the first subtraction

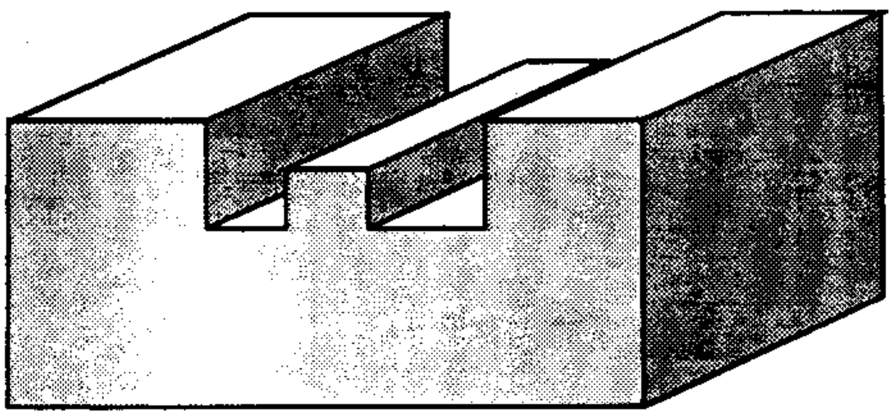


Figure 4. A part with 2 grooves which are found in the second round of recursive subdivision

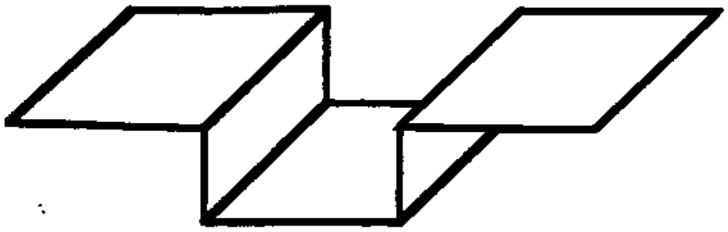


Figure 5. A groove defined by relationships between surfaces and edges

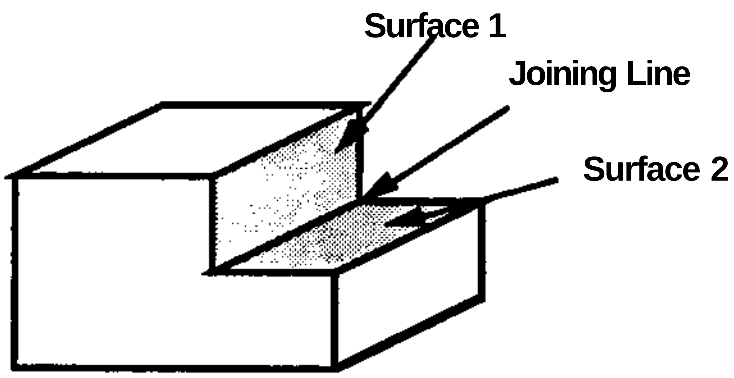


Figure 6. A shoulder as a relation between two surfaces with a common line

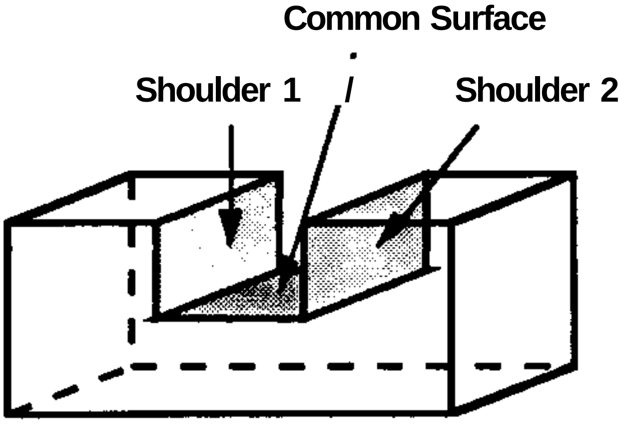


Figure 7. A relation between two shoulders and a surface defines a groove

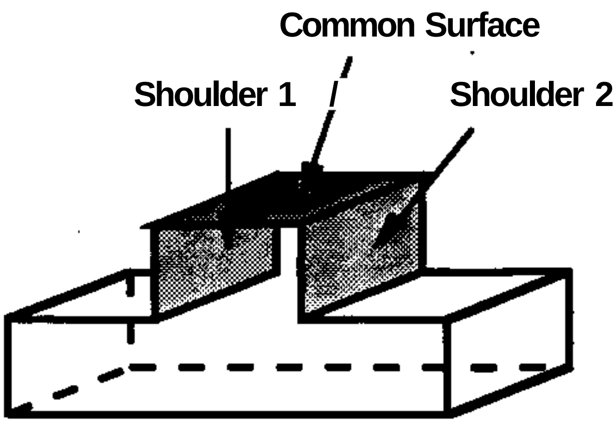


Figure 8. A relation between two shoulders and a surface defines a boss

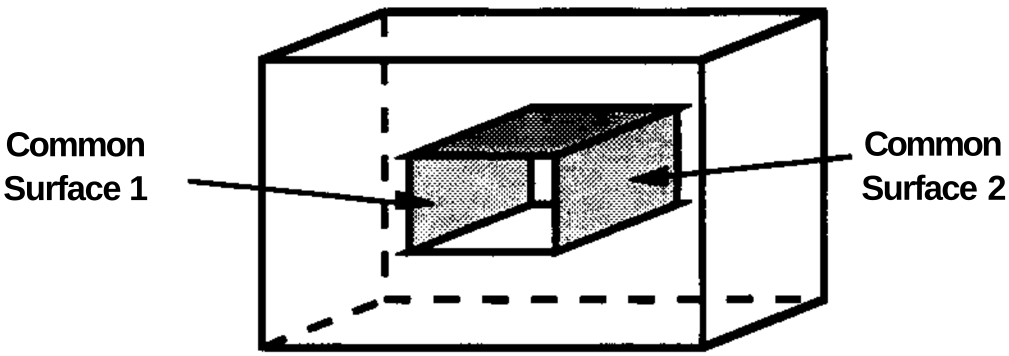


Figure 9. A hole is defined by the relation between two grooves that share a common pair of surfaces

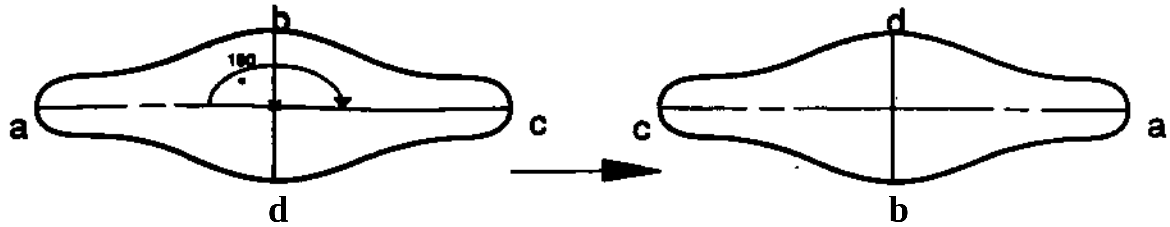


Figure lO.a Rotational Symmetry Around a Point

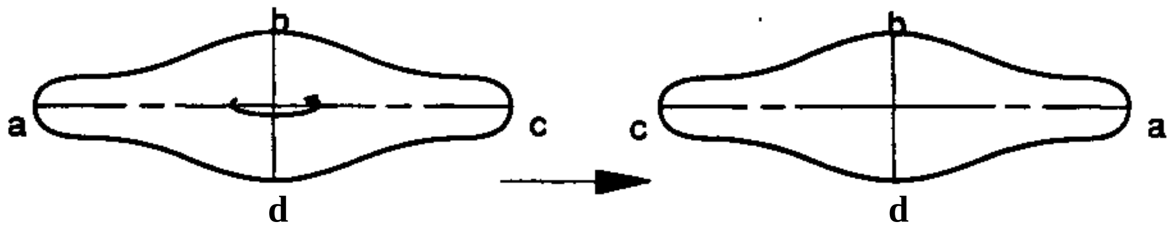


Figure lO.b Axial Symmetry around the axis b-d.

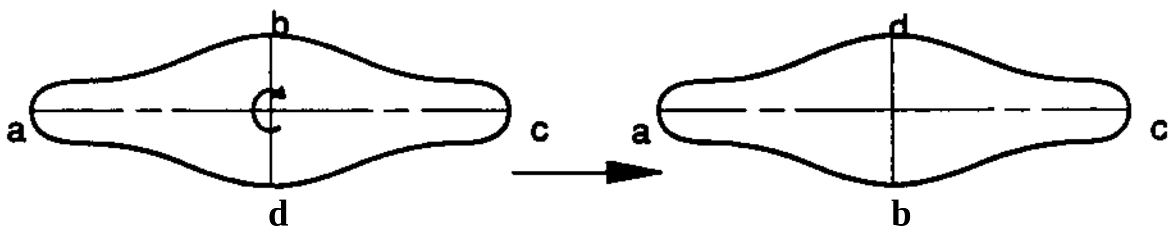
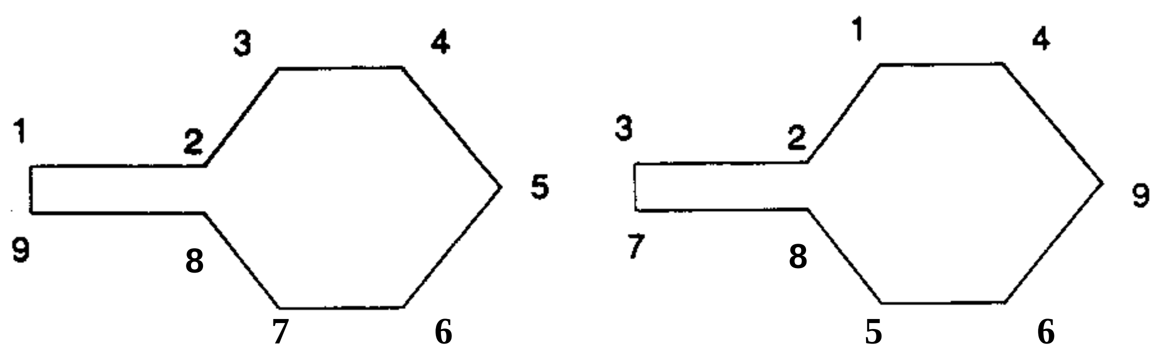


Figure lO.c Axial Symmetry around the axis a-c.



Acceptable Order Unacceptable Order

Figure 11. A figure with enumerated, ordered vertices

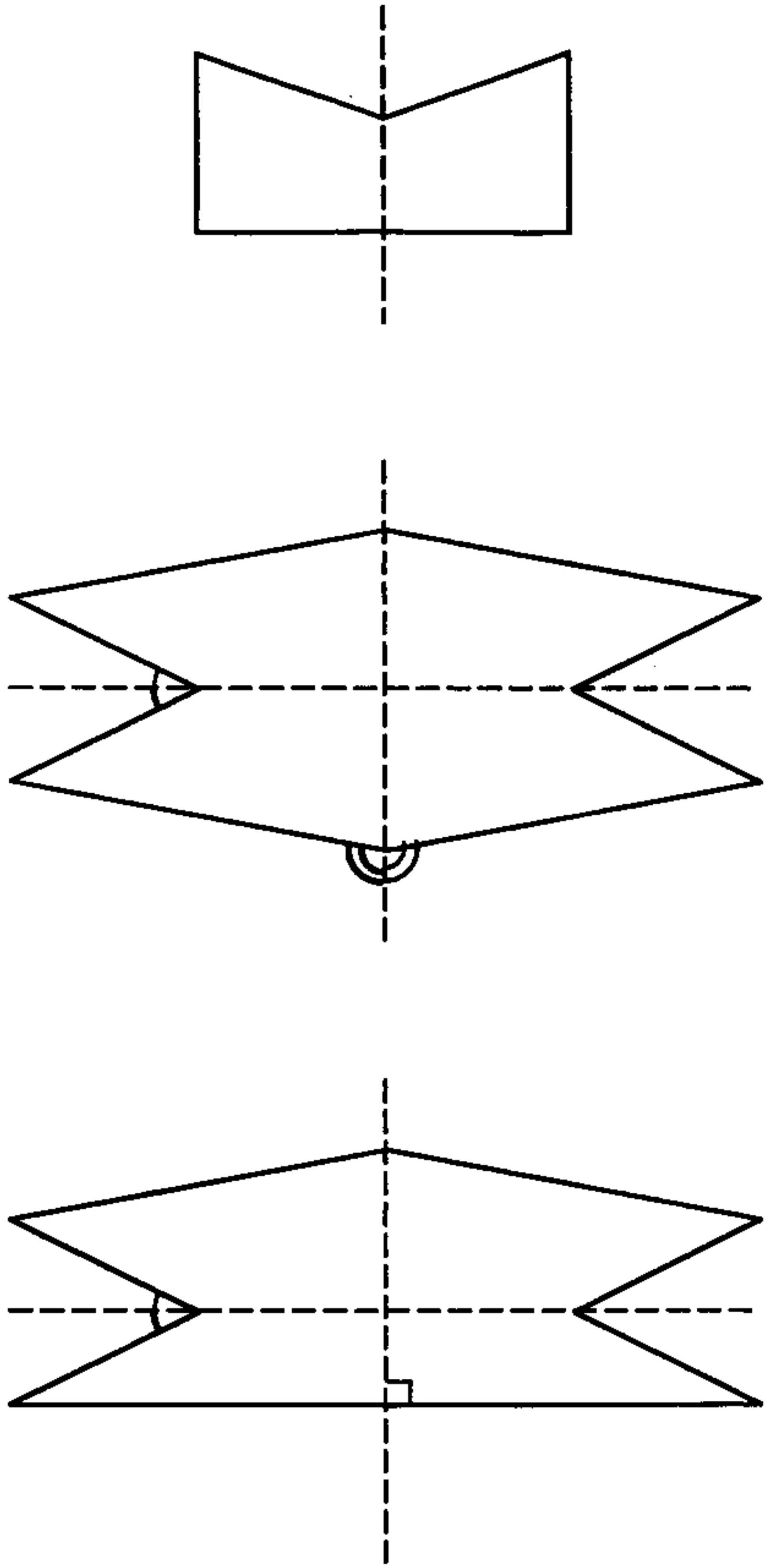


Figure 12. Symmetry axis examples

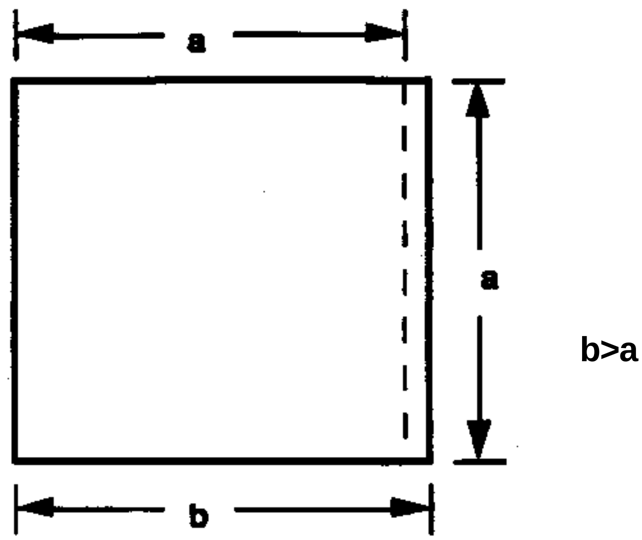


Figure 13. A nearly square object

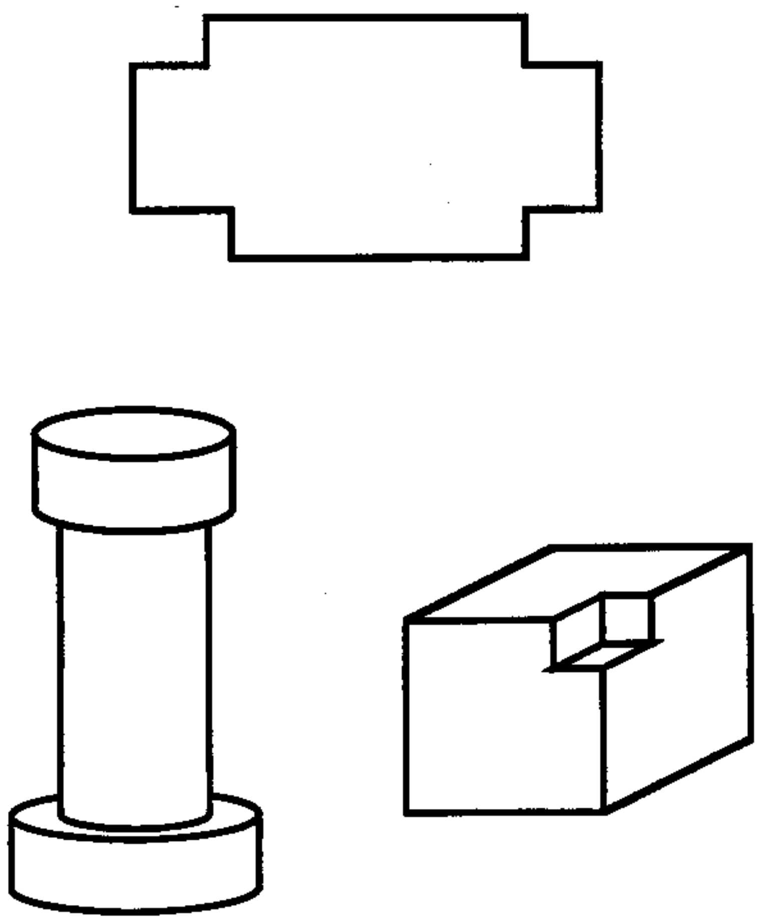


Figure 14. Nearly symmetric objects

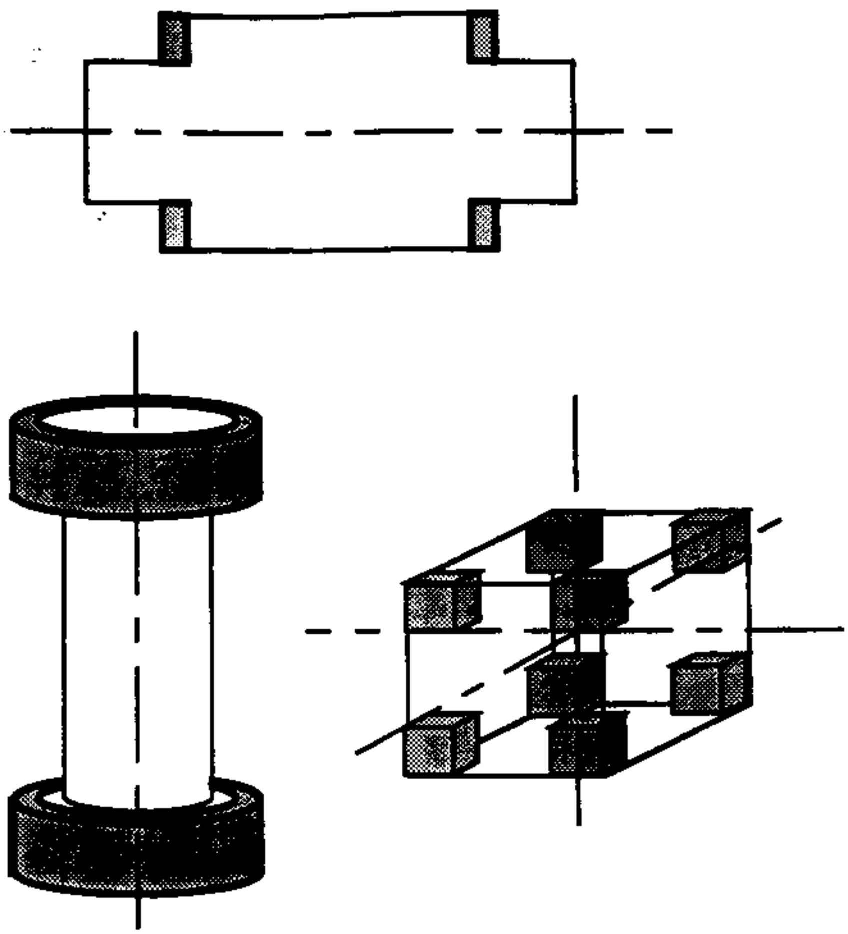


Figure 15. The difference sets for the nearly symmetric objects of Figure 14

Word count: 5198