CAR-TR-20
CS-TR-1317

DAAK70-83-K-0018
August 1983

AN ITERATIVE HOUGH PROCEDURE FOR THREE-DIMENSIONAL OBJECT RECOGNITION

Teresa M. Silberberg
Larry S. Davis
David Harwood

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

CAR-TR-20
CS-TR-1317

DAAK70-83-K-0018
August 1983

AN ITERATIVE HOUGH PROCEDURE FOR THREE-DIMENSIONAL OBJECT RECOGNITION

Teresa M. Silberberg
Larry S. Davis
David Harwood

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

This paper describes an iterative Hough procedure for recognizing images of three-dimensional objects. Straight line segments in the image are matched by finding the parameters of a viewing transformation of a three-dimensional model consisting of line segments. Assuming the scale of the object is known, there are three orientation and two translation parameters to be estimated. Initially a sparse, regular subset of parameters and transformations is evaluated for goodness-of-fit; then the procedure is repeated by successively subdividing the parameter space near current best estimates of peaks.

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAK70-83-K-0018 (DARPA Order 3206) is gratefully acknowledged.

1. Introduction

This paper is concerned with the problem of recognizing a rigid three-dimensional object in an image by matching a structural model of the object with information extracted from the image. Two important issues involved in developing such a recognition method are the choice of representation of the modeling primitives and abstractions, and the development of analysis operators that can detect instances of projections of these model entities in the image. A system is described which uses generalized Hough transforms to iteratively match a model consisting of straight edges against a set of line segments extracted from an image of that model taken from an unknown viewing position. The match is represented by a five-parameter transformation of the model onto the image. (to pA)

The overall recognition strategy in which we view the matching algorithm as being embedded would proceed by first constructing a recognition model composed of "generalized views" (i.e., continuous sets of viewpoints) that can be identified and distinguished from one another based on the visibility and ease of detection of images of the model entities. This recognition model is then used by the generalized Hough transform procedure for identifying likely instances of the object model in an image, which would finally be verified by a top-down analysis.

Section 2 describes the generalized Hough Transform matching algorithm. Examples of its application to some simple images are given in Section 3. Section 4 reviews relevant literature, and concluding remarks are presented in Section 5.

2. Representation and matching

In this section we first describe the representations on which the generalized Hough transform (GHT) matching algorithm is based, and then discuss the algorithm itself.

2.1. Representation

Object models consist of a set of three-dimensional straight line segments (edges) arranged on the surfaces of a polyhedron. Although the models have a rich intrinsic topological and geometric structure, this structure is not currently used explicitly by the matching process.

We assume orthographic projection and known scale; therefore, five parameters are needed to specify the transformation from the model to the image: three rotations and two translations. Two rotations specify the viewing transformation, and a rotation and two translations determine the orientation and position of the viewed model in the image. We use a left-handed coordinate system where rotations about the x and y axes define the viewpoint, and rotation in the image plane is about the z-axis.

The GHT algorithm constructs a "maximum likelihood" estimate of the viewing transformation using parameter space clustering techniques; the algorithm, therefore, has to have access to a five dimensional parameter space. The representation chosen for the parameter space reflects the order in which the parameters are considered by the generalized Hough transform algorithm (Section 2.2.) Viewpoints, which correspond to points on the surface of a sphere, are represented implicitly using an icosahedron, each of whose faces can be recursively divided or refined (as the GHT algorithm iterates) into four equilateral triangles. A single refinement yields eighty positions for viewpoint and

an angular resolution of approximately twenty degrees.

For each viewpoint, the image plane rotation and translations are represented explicitly using a three-dimensional array. Initially, the rotation angle is quantized to multiples of 20 degrees. The size of each translation plane for an N×N image is T×T, where T = C N / L; here C is the number of translation cells into which an average line will fall, and L is the average length of the model lines in the image. The larger C is, the finer the resolution of the space relative to the object size. It has been found through experimentation that a value of 3 for C is satisfactory. So, if the image is 128×128, average line length is 35 and C is 3, the translation space is 10×10.

2.2. The GHT matching algorithm

The method chosen for solving for the five viewing parameters is to determine the best translation and z-rotation parameters for each given viewpoint. For each viewpoint and pair of line segments, one from the model and one from the image, the model line is projected onto the image plane, rotated on the image plane to bring it parallel to the image line segment and translated to all positions in which it would "cover" the image line. The mathematical details of this transformation are presented in the Appendix; the transformation is presented graphically in Figure 1. Here, p_1p_2 is the line segment that results after the orthographic projection of the model line; i_1i_2 is the image line. The segment p_1p_2 is rotated to be parallel to i_1i_2 (q_1q_2) and translated to lie on i_1i_2. The details of the algorithm are contained in Algorithm 1; the following remarks refer to the numbered lines:

(1) Viewpoint is the set of all viewpoints as defined by the center of the faces of the

```
1   for each Viewpoint, v, do
2     HT = 0
3     for each line segment M ε Model where M is visible from v do
4       m̄=P·V·(M)
5       for each line segment m ε Image do
6         if length(m̄) ε [length(m) - k , length(m)] then do
7           compute Rot(m̄,m)
              for each θ ε Rot(m̄,m) do
8               m' = R_θ(m̄,m)
9               compute Trans(m' ,m)
                for each (x,y) ε Trans(m' ,m) do
10                HT(θ,x,y) = HT(θ,x,y) + length(m)
                end
              end
            end
          end
      end
11    update Peaks(v,θ,x,y)
    end
```

Algorithm 1

refined icosahedron.

(2) HT is the Hough Transform accumulator array in which all model-to-image matches (votes) are recorded.

(3) Model is the set of all edges in the model. In order to avoid inconsistencies that may arise if edges on opposite sides of an object are allowed to match the image simultaneously, we require that the edge be visible from viewpoint v. A "weak" model of visibility has been adopted in which a model line is visible only if the inner product of the viewpoint and the normal of the surface on which the edge lies is greater than zero.

(4) V is the viewing matrix that takes M into the viewing coordinate system. (See Appendix.) P orthogonally projects the endpoints of the line segment.

(5) Image is the set of all lines in the image.

(6) An image line and projected model line are matched only if the length of the image line is smaller than the projected model line; **k**, if small, constrains the lengths to be nearly the same. A large value of **k** supports matching a model line against a "fragmented" set of image lines.

(7) Rot($\overline{m}$,m) is the pair of angles which rotate $\overline{m}$ so that it is parallel to m.

(8) $\mathbf{R}_\theta(\overline{m},m)$ rotates $\overline{m}$ by θ so that $\overline{m}$ is parallel to m.

(9) Trans(m' ,m) is the set of x and y translation pairs that translate m' to m. Due to the fact that the line segments are not necessarily the same length, we allow all translations in which m' completely covers m.

(10) The votes in HT are weighted by the length of the image line: matching a longer line contributes a larger amount to the accumulator array. In order to allow for errors in imaging, we may also want to increment neighborhoods of the determined parameters, that is, a neighborhood or "strip" in the Hough array.

(11) After a viewpoint has been completely processed, update Peaks, a list of the five or ten parameter sets (v,θ,x,y) which have the highest number of votes.

Algorithm 1 provides an ordered list of estimates of likely viewing transformations of the model using a coarsely quantized Hough space. Better estimates are now obtained by successively refining the Hough space as long as well-defined peaks can be detected. During this refinement process, the resolution of each parameter is increased. Viewpoint resolution is doubled by dividing each face of the current icosahedron-based structure

into four faces. The resolution of the image plane rotation and translation parameters are doubled by halving each current interval.

The algorithm presented below is similar to Algorithm 1 with the exception of two additional features. First, only those parts of the Hough parameter space in the neighborhoods of the estimated peaks are refined; that is, only those refined viewpoints that are derived from previously estimated viewpoints are considered, and only those image plane rotations and translations that fall within a small interval about the previously estimated parameters are considered. Second, Algorithm 2 is iterative and continues until some halting criterion is met. After some number of iterations, we find that the Hough technique no longer provides useful results since the votes are eventually randomly scattered in the over-quantized parameter space. When this occurs, the iterations are halted and the previous peak is the final estimate of the transformation.

The details are provided in Algorithm 2; the following remarks apply to the numbered lines:

```
1   for each (v,θ,x,y) ∈ Peaks do
2      compute Refinement(v)
3      compute Interval(θ,x,y)
       for each r ∈ Refinement(v) do
4         compute HT'
       end
       update Peaks
    end
    if halting criterion not met, begin Algorithm 2 again.
```

Algorithm 2

---

(1) In the first iteration, Peaks is the list of parameters computed by Algorithm 1.

(2) Refinement(v) is the set of viewpoints resulting from the refinement of view v; in each successive iteration, smaller neighborhoods around v are considered.

(3) Interval(θ, x, y) corresponds to the intervals of these parameters over which further refinement in the Hough space is considered.

(4) Computation of the final rotation and translation parameters is similar to that of Algorithm 1 except for the interval restrictions discussed above.

This algorithm iteratively estimates, with increasing precision, an object's orientation and location in an image. The precision is ultimately limited by the model and imaging accuracy, and the fineness of the image features.

3. Examples

The initial estimation and refinement steps have been used to compute the orientation and position of a three-dimensional rectangular solid with surface markings from an image. Examples of views of the object are shown in Figure 2. Figure 3 is a rough sketch where the letters X, M and L are opposite the letters K, T and H, respectively. The model features extracted are the edges that the letters make with the surfaces upon which they lie. The model consists of pairs of endpoints for each edge along with the normal of the surface containing it. These are the surface normals used to determine visibility from a given viewpoint.

The image data consists of straight edges found in Figure 2. The edge pictures are shown in Figure 4. These edges are found by first applying a 3×3 Laplacian to the smoothed image. This is followed by a thinning operation and an algorithm that allows only 2-connected chains to remain. Endpoints of straight edges are then extracted.

The initial viewpoint structure in this example is a once refined icosahedron that has 80 faces. The viewpoints, resolved at approximately 20 degrees, are in the centers of the faces. Figure 5 displays a representation of the icosahedron. The left and right sides of the figure wrap around, and the letters correspond to vertices. The heavy lines surround the faces of the original icosahedron, and the dashed lines represent the additional faces after refinement.

The resolution of the image plane rotation is 20 degrees. In the example, the average length of a model edge in the image is 40; the image size is 128×128. If we choose $C = 3$ and use $T = C\,N / L$, the translation space is 10×10. We actually begin

with a space that is 8×8 (C = 2.5) and has a resolution of 16 pixels.

An examination of Figure 4 shows that no more than half of any object line is missing from the image (consider the L), so k= 20 or half of the average length of a model edge. We include a heuristic that does not allow isolated noise lines to be considered. We require that for each line in the image, there is another line within a minimum distance of it. This distance is computed from the maximum distance between any two edges in the model.

At the conclusion of Algorithm 1, a list of the 10 most likely parameter sets is in Peaks. Henceforth, TK refers to the processing of the image with a T and a K; likewise, TKL refers to the processing of the image with T, K, and L. The distribution of the viewpoints for TK ranked by the number of votes is shown in Figure 6a; viewpoints for TKL are shown in Figure 6b. The votes of the top five parameter sets are:

TK	TKL
248	214
211	179
183	179
169	173
163	163

Algorithm 2 begins by doubling the resolution of each parameter, but only within the neighborhoods of the peaks. Two iterations of this procedure yield a resolution of 5 degrees for the three rotation parameters and 4 pixels for the translation parameters. Figure 7 shows the distribution of viewpoints after the first iteration. The angular distance is now 10 degrees. Notice that in Figure 7a, the highest 6 viewpoints are

clustered within a 15 degree radius. The votes of the top five peaks are as follows:

TK	126	TKL	105
	120		105
	119		104
	118		101
	109		100

Figure 8 shows the top 5 peaks of the last iteration. The voting is as follows:

TK	122	TKL	107
	119		87
	92		80
	89		77
	88		71

Notice that TK's viewpoints are clustered more in Figure 8a, but the highest peak in TKL is very strong.

If the algorithm iterates a third time, the results do not improve. Because of errors in image digitization and edge extraction, a very fine quantization causes a decrease in the number of actual matches. Additionally, the voting that results from any matches becomes scattered among neighboring parameters so that there is no strong peak.

Figures 9 (for TK) and 10 (for TKL) show the projection of the model using the best peaks for TK and TKL. Figure 9a is the projection using the computed orientation and translation; Figure 9b superimposes the model projection onto the original image. Figure 10 is similar.

4. Relevant Literature

The major issues addressed in this paper involve representation of the model primitives and abstractions and development of the recognition algorithm.

Observed model features can be directly related to viewing position; Koenderink and van Doorn [Koen79] and Chakravarty and Freeman [Chak82] utilize this property in their formulations of "aspect graphs" and "vantage-point domains", respectively. This kind of representation is compatible with the GHT method that we have presented and could be used to provide an initial set of representative viewpoints.

In this paper, the generalized Hough transform [Davi82, Ball81] was used as a clustering algorithm in which recognition of image and model features is followed by determination of a transformation that maps the model features to the image features. This algorithm is similar to one used by Stockman [Stoc82] who describes a method that matches two-dimensional edge data in problems of image registration and two-dimensional object recognition.

We have represented the Hough parameter space as a three-dimensional array for the image plane rotation and translation parameters, leaving the other two dimensions of viewpoint parameters implicit. Several space-efficient representation schemes for multidimensional Hough spaces have been suggested. These include hash tables, or cache accumulators [Brow82], and dynamically quantized data structures such as quadtrees [Sloa81] and k-d trees [O'Rou81].

The GHT procedure discussed in this paper can be embedded in a more general system that iteratively detects image features, predicts the occurrence of the object model based on the detected features, and verifies the prediction using a top-down procedure. Such a technique has been used by Roberts [Robe65] in a system that recognizes objects composed of cubes, wedges and prisms in a line drawing, and by Brooks [Broo83] in a system that identifies instances, orientations and positions of three-dimensional objects.

5. Concluding Remarks

The system described in this report provides a basis for recognizing objects in an efficient manner. The generalized Hough transform is resistant to noise and can match based on only partial information. The majority of the computation is in estimating orientation and position parameters using a low resolution, coarsely quantized Hough space and matching only highly expected features. Not until a likely viewpoint is found does the system require and use more specific information. The system can be extended to include a preprocessing stage in which view-dependent model features are identified and organized into a recognition model, and a final refinement stage in which the estimated transformation is verified and strengthened using a top-down process.

APPENDIX

A.1 Viewpoint transformation

Using a left-handed coordinate system, the model is represented as viewed from (0, 0, -1). To determine the viewing coordinates from a viewpoint (a, b, c) lying on a unit sphere, we compute the transformation that takes (a, b, c) to (0, 0, -1). See Figure 11.

First rotate (a, b, c) about the y-axis so that it lies in the y-z plane. The rotation matrix for angle θ_y is

$$\begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \text{ or}$$

$$R_{\theta_y} = \begin{bmatrix} \frac{-c}{\sqrt{1-b^2}} & 0 & \frac{-ba}{\sqrt{1-b^2}} \\ 0 & 1 & 0 \\ \frac{a}{\sqrt{1-b^2}} & 0 & \frac{-c}{\sqrt{1-b^2}} \end{bmatrix}$$

Next rotate (a_{θ_y}, b, c_{θ_y}) about the x-axis so that it lies on (0, 0, -1). The rotation matrix for θ_z is

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_z) & \sin(\theta_z) \\ 0 & -\sin(\theta_z) & \cos(\theta_z) \end{bmatrix} \text{ or}$$

$$R_{\theta_z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{1-b^2} & b \\ 0 & -b & \sqrt{1-b^2} \end{bmatrix}$$

The desired viewing transformation, T, is

$$T = R_{\ell_y} \times R_{\ell_x} = \begin{bmatrix} \frac{-c}{\sqrt{1-b^2}} & \frac{-ab}{\sqrt{1-b^2}} & -a \\ 0 & \sqrt{1-b^2} & -b \\ \frac{a}{\sqrt{1-b^2}} & \frac{bc}{\sqrt{1-b^2}} & -c \end{bmatrix}$$

This transformation has the property of keeping points in the upper (lower) part of the viewed object in the upper (lower) part of the transformed object; consequently, special precautions must be taken for viewing positions equal to (0, 1, 0) and (0, -1, 0).

A.2 Rotation in the image plane

We want to rotate a 2d line segment M to be parallel to a segment I. There are two such rotations. We represent M by its endpoints (A,B) and I by its endpoints (C,D). Figure 12 shows the unit vectors M_u and I_u in the direction of the segments M and I. The problem is determination of the angle $\gamma=\beta-\theta$ between M_u and I_u. In polar coordinates, $M_u=(\cos\theta,\sin\theta)$ and $I_u=(\cos\beta,\sin\beta)$. Using the trigonometric identities for addition of angles,

$\cos(\gamma)=\cos(\beta-\theta)=\cos(\beta)\cos(\theta) + \sin(\beta)\sin(\theta)$,

$\sin(\gamma)=\sin(\beta-\theta)=\sin(\beta)\cos(\theta) - \cos(\beta)\sin(\theta)$, or

$\cos(\gamma)=i1\ m1 + i2\ m2$,

$\sin(\gamma)=i1\ m1 - i1\ m2$.

This uniquely determines γ. The second angle is $\gamma+\pi$.

A.3 Translation in the image plane

We want to compute all discrete translations that take a line M with endpoints (A,B) to a line I with endpoints (C,D) such that M covers all of I. (See Figure 13.) To do this, compute all translations that cause M to "slide" along I a unit distance. First compute the translation line T with endpoints (X,Y) = (C-A,D-B). Every point on this

line segment defines a translation from M to I. The desired discrete translations are $X + i \cdot (X_u, Y_u)$ for $i = 0$ to $Y-X$, where $(X_u, Y_u) = \frac{(Y-X)}{Y-X}$ is the unit vector in the direction of T. Figure 15 shows an example.

$M=((1,-1),(1,5))=(A,B)$

$I=((7,4),(7,8))=(C,D)$

$T=((6,5),(6,3))=(X,Y)$

$(X_u, Y_u)=(0,-1)$

$Y-X=2.$

The translations are:

$(6,5)+i \cdot (0,-1)$ for $i=0$ to 2, or

$(6,5)+0 \cdot (0,-1)=(6,5)$

$(6,5)+1 \cdot (0,-1)=(6,4)$

$(6,5)+2 \cdot (0,-1)=(6,3).$

References

[Ball81]

D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recognition*, volume 13, 111-222, 1981.

[Broo83]

R.A. Brooks, "Model-based three-dimensional interpretations of two-dimensionl images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, 140-150, 1983.

[Brow82]

C.M. Brown, D.B. Sher, "Hough transformation into cache accumulators: considerations and simulations", TR114, Computer Science Department, University of Rochester, 1982.

[Chak82]

I. Chakravarty, H. Freeman, "Characteristic views as a basis for three-dimensional object recognition", *Proceedings, SPIE*, volume 336, 37-45, 1982.

[Davi82]

L.S. Davis, "Hierarchical generalized Hough transforms and line-segment based generalized Hough transforms", *Pattern Recognition*, volume 15, 277-285, 1982.

[Koen79]

J.J. Koenderink, A.J. van Doorn, "The internal representation of solid shape with respect to vision", *Biological Cybernetics*, volume 32, 211-219, 1979.

[O'Rou81]

J. O'Rourke, "Dynamically quantized spaces for focusing the Hough transform", *Proceedings, IJCAI-7*, Vancouver, B.C., 737-739, 1981.

[Robe65]

L.G. Roberts, "Machine perception of three-dimensional solids", *Optical and Electro-optical Information Processing*, J.T. Tippett et. al., eds., MIT Press, Cambridge, MA, 159-197, 1965.

[Sloa81]

K. Sloan, "Dynamically quantized pyramids", *Proceedings, IJCAI-7*, Vancouver, B.C., 734-736, 1981.

[Stoc82]

G. Stockman, S. Kopstein, S. Benett, "Matching images to models for registration and object detection via clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 4, 229-241, 1982.

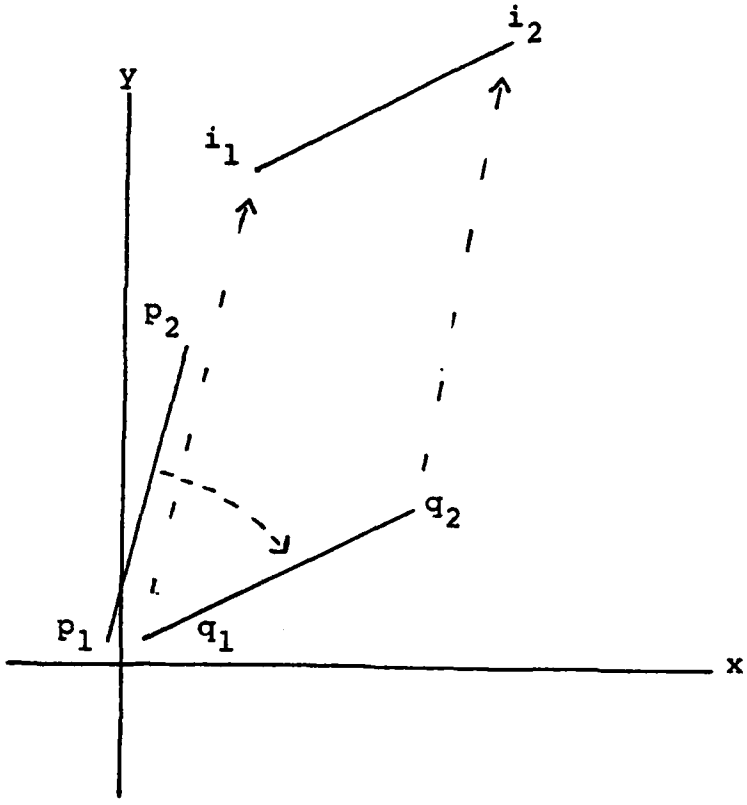


Figure 1

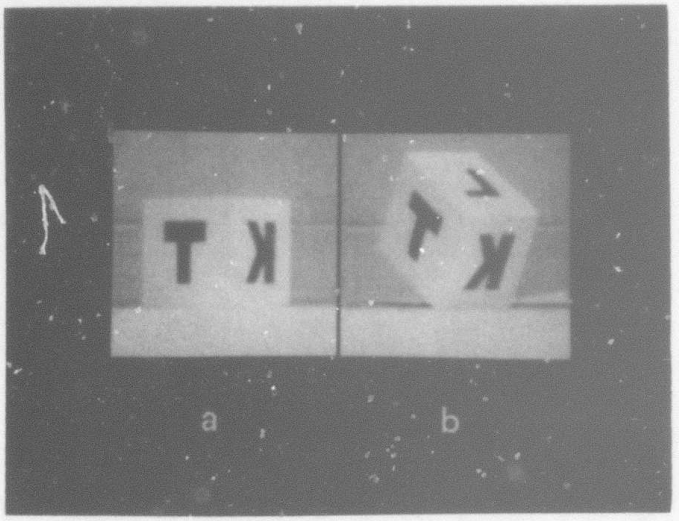


Figure 2

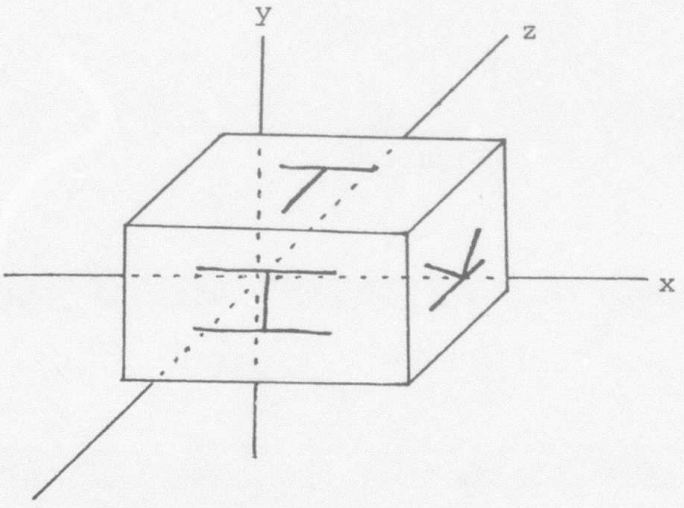


Figure 3

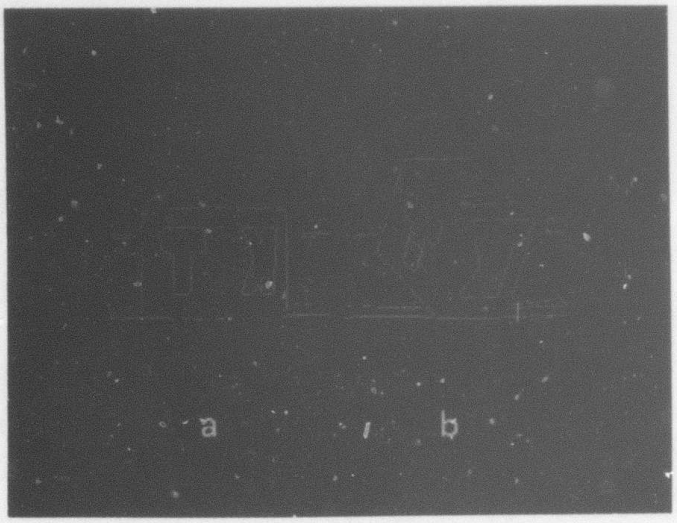


Figure 4

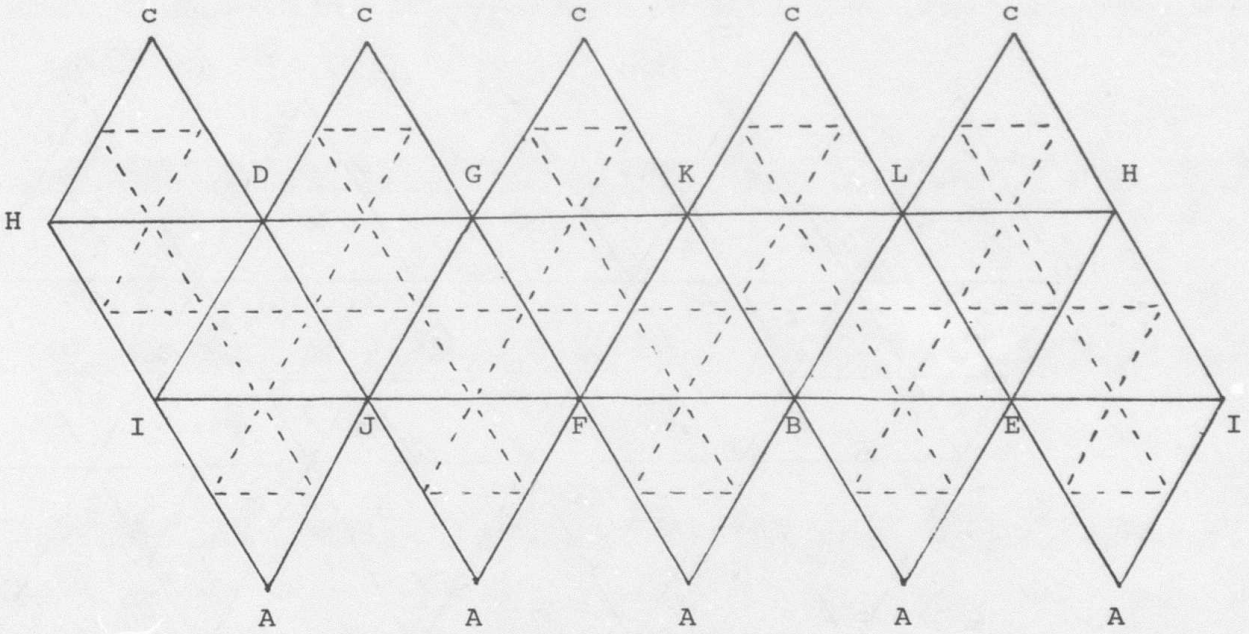
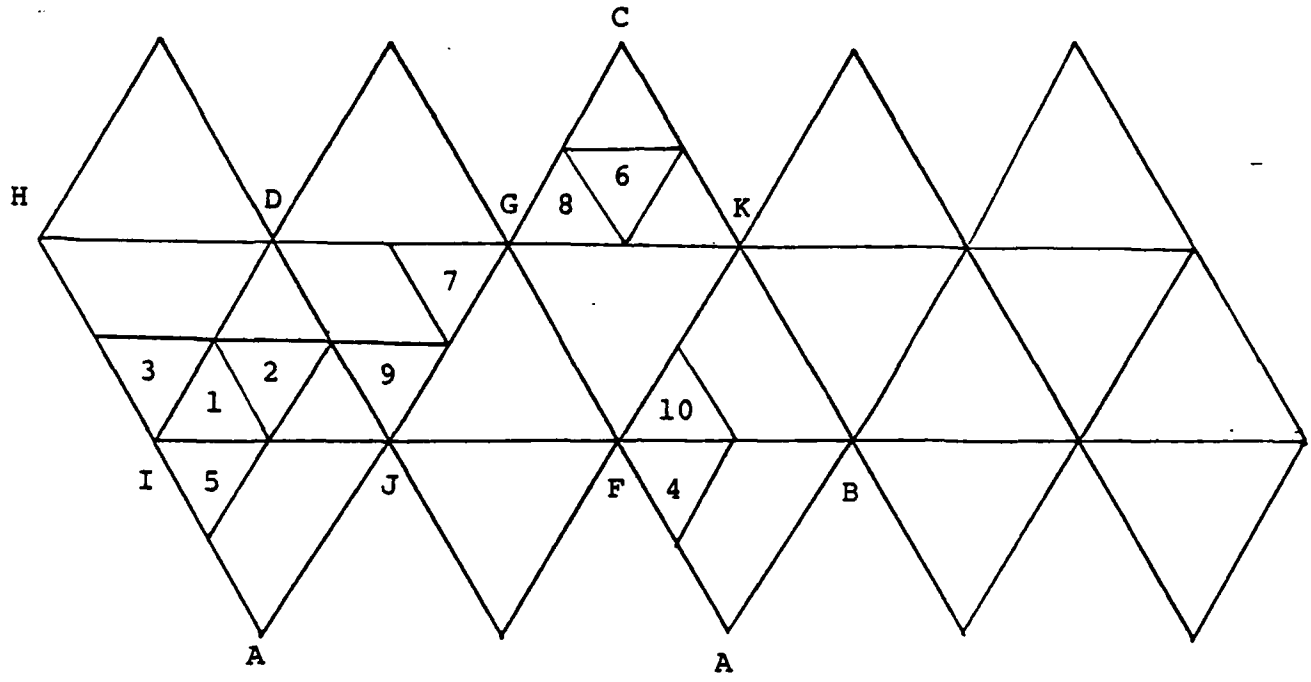
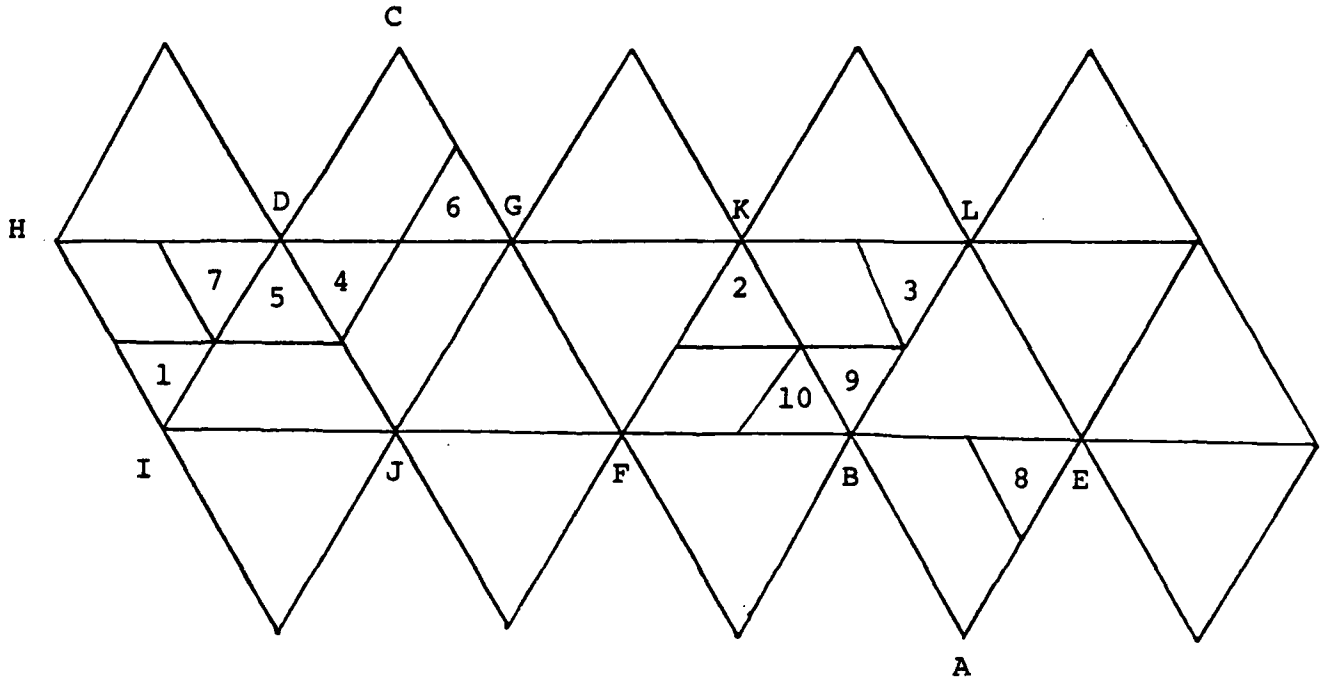


Figure 5

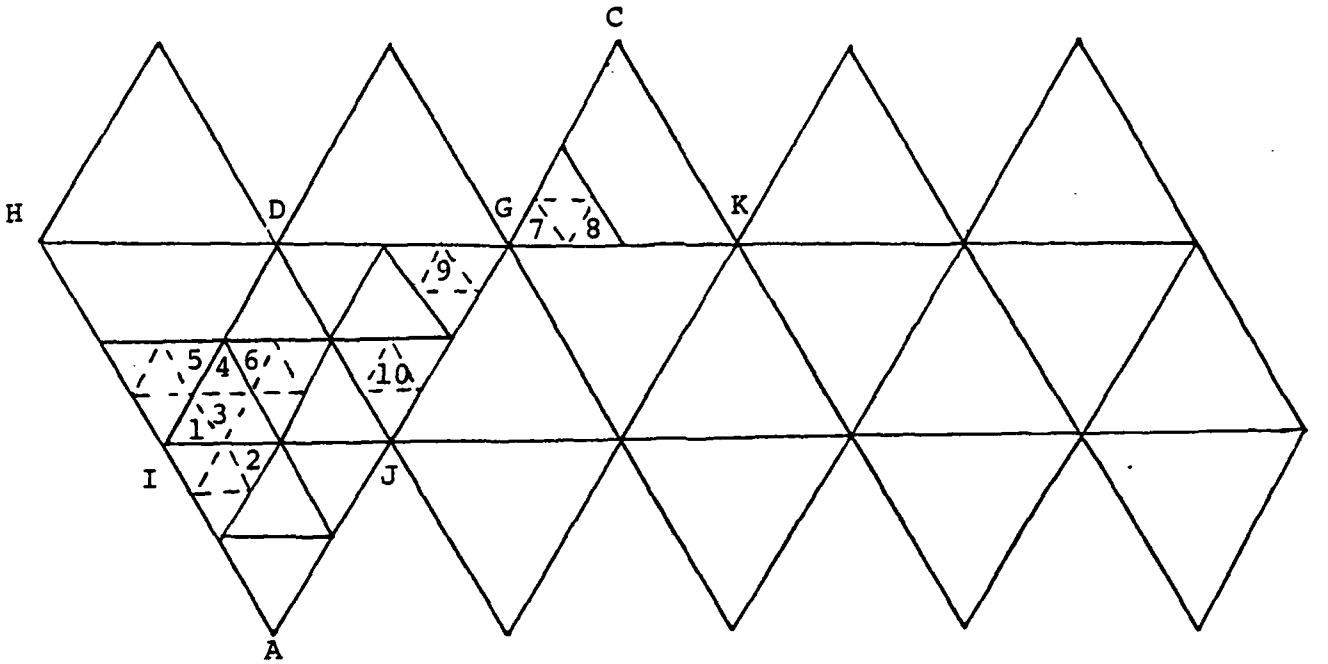


a.

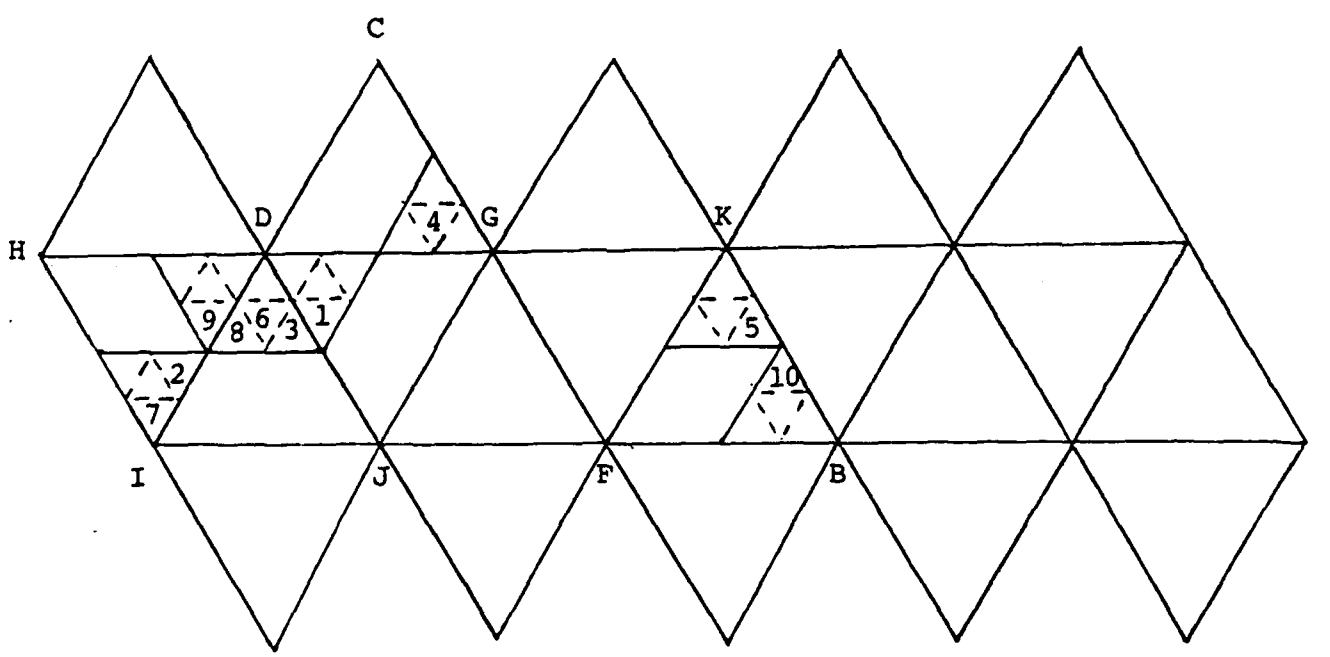


b.

Figure 6

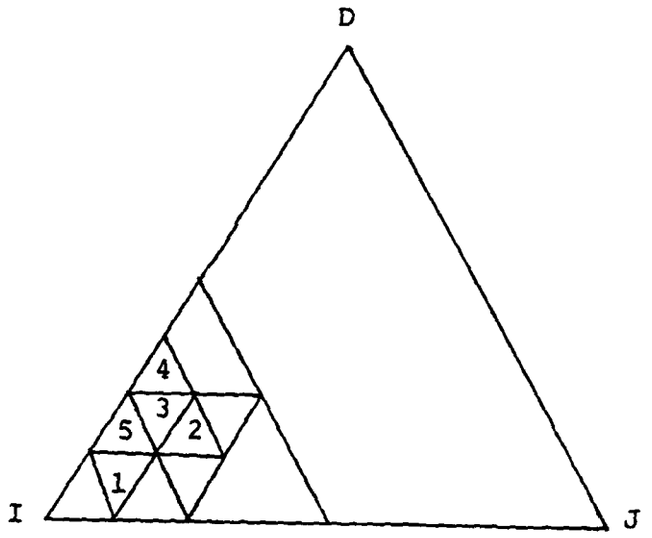


a.

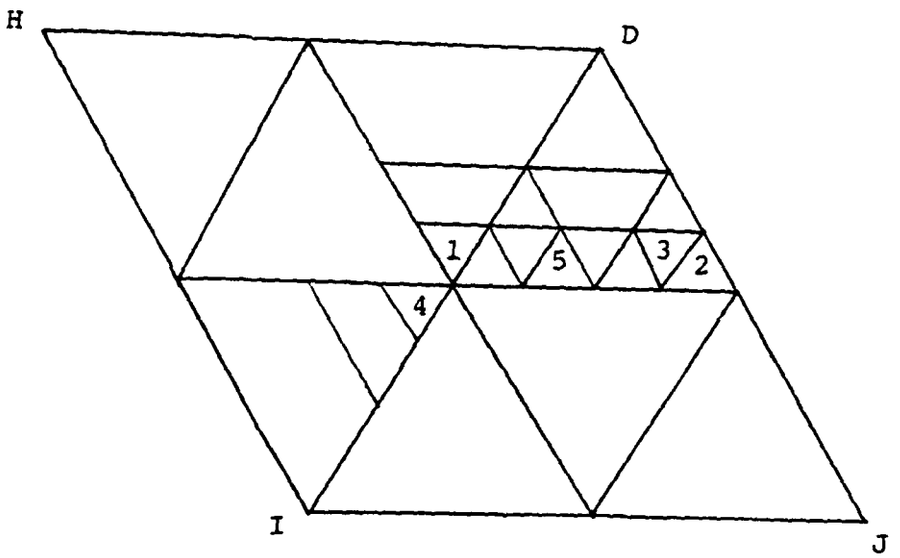


b.

Figure 7



a.



b.

Figure 8

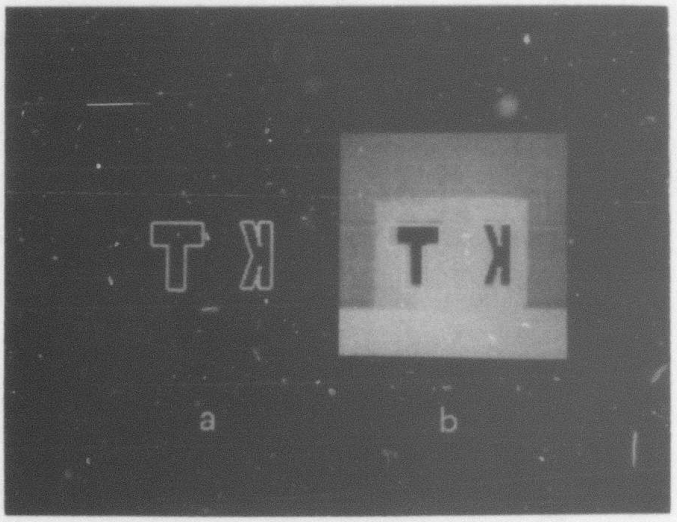


Figure 9

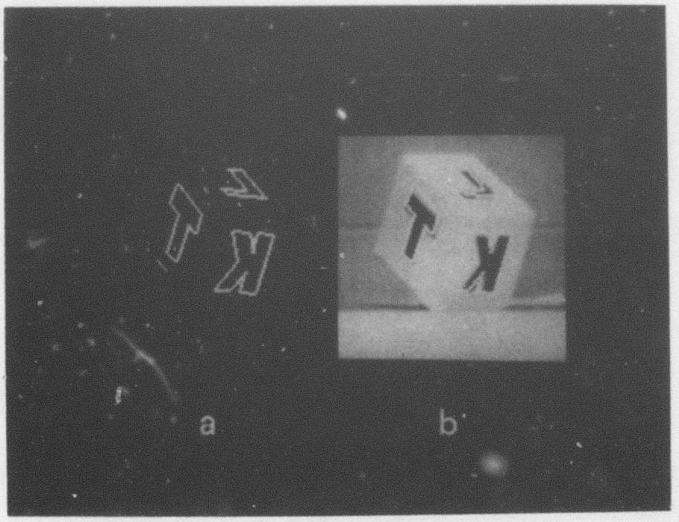


Figure 10

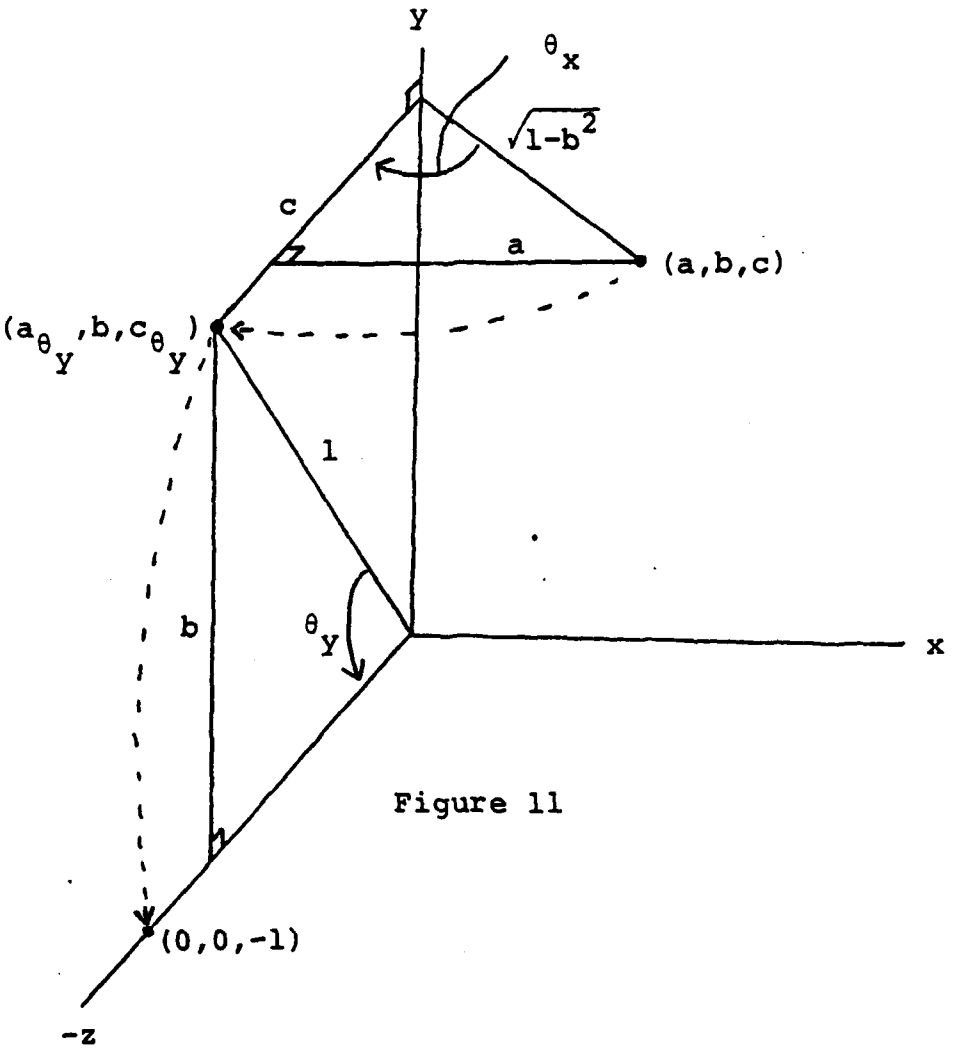


Figure 11

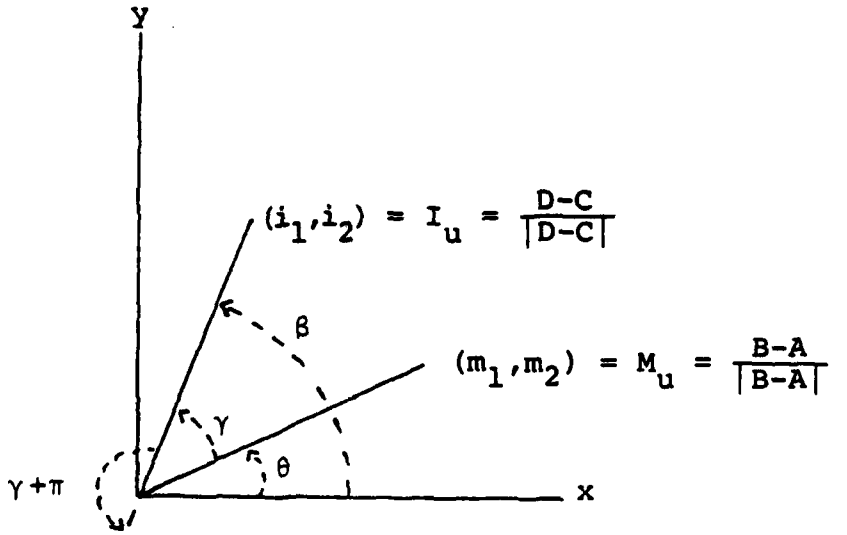


Figure 12

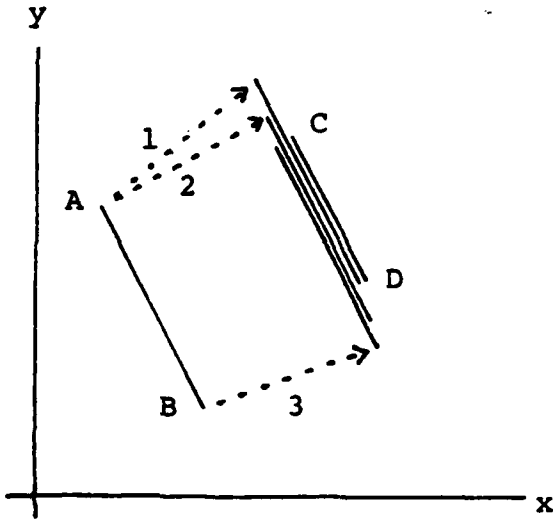


Figure 13

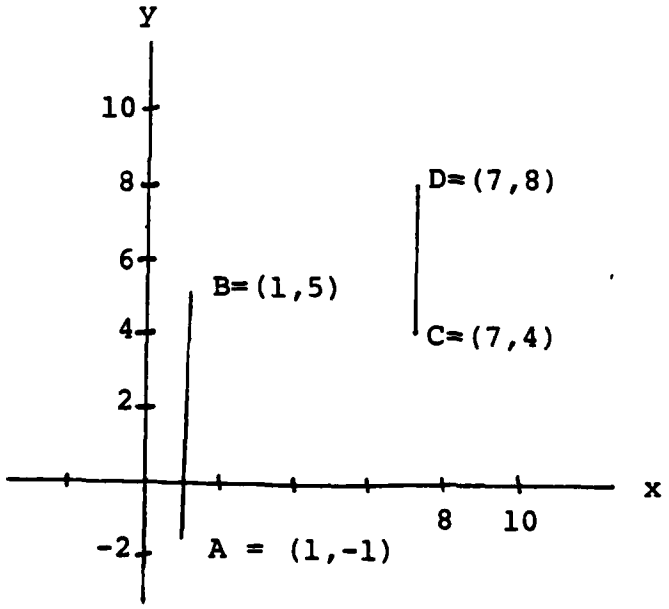


Figure 14