A Technique for 3-D Robot Vision for Space Applications[1]

V. Markandey, H. Tagare, and R.J.P. deFigueiredo
Rice University
Houston, TX 77251-1892

1) ABSTRACT:

This paper reports an extension of the MIAG algorithm for recognition and motion parameter determination of general 3D polyhedral objects based on model matching techniques and using *Moment Invariants* as features of object representation. Results of tests conducted on the algorithm under conditions simulating space conditions are presented.

2) INTRODUCTION:

Many different object recognition and attitude determination techniques have been proposed by researchers. The earliest ones used the approach of matching the observed image to a library of a fixed number of views of objects. The limitations of such an approach are glaringly apparent. Among the later techniques, Richard and Hemami [1] used Fourier descriptors and Dudani et al [2] used moment invariants. Watson and Shapiro [3] used a model matching technique to identify wireframe perspective views. Their method is iterative and requires use of a numerical optimization technique. Marr and Poggio [4] have implemented a stereo reconstruction algorithm which uses geometric constraints to recover surface shape. Similar range data techniques have been developed by other researchers also. A fundamental limitation of these techniques is the introduction of restrictive assumptions about the imaged scene in terms of generalized cones [5] or in terms of planar and quadric patches [6]. Horn [7] has worked on the extraction of shape from shading, using the reflectance map. This method uses the brightness gradient as the image feature used in recovery. It is applicable to smooth, uniform Lambertian surfaces. Stevens [8], Kender [9] and later Witkin [10] have tried to recover shape from texture. This technique and also the shape from contour (surface boundaries) technique presented by Barrow and Tenenbaum [11] rely on the assumption that the world of objects is regular. Such techniques are limited to smooth-textured surfaces. For some other contributions see Silcox [12].

Bamieh and deFigueiredo [12] have developed the Moment-Invariants / Attributed Graph (MIAG) Algorithm in which 2D moment invariants which are invariant under 3D motion, have been used for the recognition of 3D objects, using an attributed graph representation and based on the concept of model matching. This approach avoids restrictive geometric assumptions and so offers an advantage over most techniques discussed above. In its original form this algorithm was applicable for recognition of polyhedral objects but it could not be used for attitude determination if the polyhedron had symmetric faces for reasons discussed later in this paper. This limitation has been overcome now as discussed in this paper. As this technique uses moment invariants as features of representation and these can be computed only for planar faces, the technique is not directly applicable to the recognition and attitude determination of curved objects. However we can use other representational features such as the Gaussian and mean curvature and the attributed graph and model matching techniques can still be applied.

To implement this technique we need picture information in the form of wireframes. So the picture from the camera is digitized and converted to wireframe form before applying the MIAG algorithm to it. In the work currently in progress, the overall process is divided into three parts:

[1]Supported by the NASA contract NAS 9-17145, the NASA/JSC grant NCC 9-16, and the NSF grant DCR-8318514.

1) Data acquisition and digitization.

2) Wireframe extraction.

3) Recognition and motion parameter determination.

The first two parts above constitute the image processing/feature extraction stage. The work in this stage is briefly outlined below.

3) IMAGE PROCESSING / FEATURE EXTRACTION:

3.1) Data acquisition and digitization: Models of various space objects, such as mockups of satellites, the space shuttle and parts of the space station are being used to test the performance of the algorithm. These models are grabbed by cameras under illumination conditions simulating those prevelent in space orbit. The pictures are digitized to obtain 2D arrays of brightness values. This is the initial level of representation in the system.

3.2) Wireframe extraction: Wireframe extraction consists of removing noise from the picture and subjecting it to edge detection and reconstruction. The input image is lowpass filtered to remove the high frequency noise. A 7x7 Gaussian filter is used for this. The Sobel gradient operator is then applied to the output of the lowpass filter to obtain an edge detected version of the input image. This image is a grey level image. It is converted to binary form by thresholding, which also removes some of the noise and thins down the edges. The remaining noise is removed by median filtering. A length 5 filter was employed for this. The output so obtained is a noise free, binary edge image. But the edges are thick smears instead of the fine lines required in a wireframe. A thinning algorithm [13] is applied to this to reduce the edges to unit pixel thickness, thus obtaining the required wireframe.

4) RECOGNITION AND MOTION PARAMETER DETERMINATION:

The MIAG algorithm [12] is an algorithm of recognition and attitude determination of 3D objects. We discuss this algorithm in two steps: First, object recognition and second, attitude determination.

4.1) Object recognition: The MIAG technique recognizes a 3D object from its projection on an imaging plane. The algorithm works for the identification of polyhedral objects. Each face of a polyhedron can be considered as a rigid planar patch (RPP). Motion of the object can then be considered as motion of its constituent RPP's. If it is assumed that the image is formed by parallel projection then if an RPP undergoes rigid body motion in 3D its image undergoes affine transformations. So the method which tries to identify an object in 3D motion should use features of images which remain invariant under affine transformations. General moment invariants are such features. They remain invariant under translation, rotation and scale changing. Moments are coefficients in a series expansion of the image function, similar to those in a Fourier series expansion. But unlike in Fourier series where sine and cosine functions are the basis functions, here the basis functions are polynomials in the image function variables. Thus if the picture function is f(x,y) its moment is:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy$$

for p,q=0,1,2...

The value of (p+q) is known as the order of the moment. Theoretically, for a perfect description of the picture in terms of moments, p and q should go to ∞. But in the present algorithm moments upto only order four have been used. This is because the computation of higher order moments is increasingly difficult, and it was found that picture representation in terms of four moments gives good results in the test cases.

These moments have to be computed for each face of the picture wireframe. The picture intensity is taken to be 1 inside a polygon and 0 outside it. Thus all the picture information is contained in its boundaries. Using this fact, the above surface integral can be changed to a line integral by Green's theorem. For a digital picture the integral reduces to a sum. See [12] for details.

Moment invariants of all faces of certain standard objects are stored in the system library. Given a wireframe which needs to be identified, the moment invariants of each of its faces are computed. These are then matched to the stored values of the moment invariants of the library objects. If all the moments corresponding to a face of the RPP match all the moments of a face of a stored object, we can say that the two faces are similar. For the objects to be the same, not only should the faces be similar but the adjacency

conditions of the faces and the angles between the faces should be similar. To carry out this matching, the wireframe is first converted to an attributed graph. Each node of the graph represents a face of the wireframe. If two nodes are connected by a line (edge) it means that the faces corresponding to these nodes are adjacent. With each node is associated a feature vector consisting of a set of moment invariants of the face that it represents and with each edge is associated a scalar which gives the angle between normals to the two faces it connects. As an example, the attributed graph representation of a cube is shown in Fig 1.

The algorithm works as follows: Suppose we hypothesize that node W_j in the wireframe corresponds to node O_j in the model graph. If W_j has k nodes adjacent to it, $W_{m1},...,W_{mk}$ then O_j should also have k nodes adjacent to it, $O_{n1},...,O_{nk}$. The following constraints should be satisfied:

1) W_{ms} must have the same feature vectors as O_{ns}, s=1,...,k.

2) Angle between W_{ms} and W_j should equal angle between O_{ns} and O_j for all s=1,...,k.

3) If any two W_{ms} 's are connected then the angle between them should equal that between their matching nodes.

If all these conditions are satisfied, then an admissible matching configuration is said to have been obtained at nodes W_j and O_j. If matching configurations are obtained between all nodes of the given wireframe and one of the stored models, then we can say that the wireframe matches the model. In most cases, only a small part of the given model needs to be matched to discriminate it against the other models in the library.

It may be noted that because of numerical truncations and rounding during calculations there may not be a perfect match between the moment invariants computed for the wireframe and those stored for the model. So we define a measure of error between the two sets of moment invariants. The moment invariants can be taken as coordinates of a point in four dimensioinal vector space and the distance between the two points is taken as a measure of error. If I_1, I_2, I_3 and I_4 are the moment invariants of a wireframe's face and I'_1, I'_2, I'_3 and I_4 those of a model's face then the distance is:

$$d = \sqrt{(I_1 - I'_1)^2\rho_1^2 + (I_2 - I'_2)^2\rho_2^2 + (I_3 - I'_3)^2\rho_3^2 + (I_4 - I'_4)^2\rho_4^2}$$

where the ρ's are weighting factors. These are needed to equalize the contribution of all four moment invariants in the error measure because some of the moment invariants may have values of the order of 10^{-3} and others may be of the order of 10^{-7}. If the value 'd' is less than a certain threshold (taken 0.01 here), the two sets of moment invariants are taken to be equivalent.

The driver algorithm arbitrarily picks a node W_j in the wireframe, then it looks for a node O_j in the model with the same feature vector. If matched, these nodes are marked as a pair. An adjacent image face is chosen and the adjacent object faces are scanned to see if one of them matches it. As each adjacent pair is found it is checked for consistent adjacency and equality of angles between faces. If everything matches satisfactorily a succesful match is declared.

4.2) Attitude determination: The identity of the object having been so determined, one has to estimate the attitude and location of the recognized object relative to a library standard.

Let (X,Y,Z) be the original coordinates of a point on the body and (X',Y',Z') be its coordinates after motion. Then,

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T$$

where

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_8 \end{bmatrix}$$

is the rotation matrix and

$$T = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}$$

is the translation matrix.

Let the corresponding points on the image be (x,y) and (x',y'). Then,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r_1 x + r_2 y + r_3 Z + \Delta x \\ r_4 x + r_5 y + r_6 Z + \Delta y \end{bmatrix}$$

For simplicity let Z=0 i.e. the RPP in library lies in the x-y plane of the object space. Then the above matrix can be repressented as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Q_1^1 & Q_2^1 \\ Q_1^2 & Q_2^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}$$

where

$Q_1^1 = r_1, Q_2^1 = r_2, Q_1^2 = r_4, Q_2^2 = r_5, \Delta x = \Delta X, \Delta y = \Delta Y.$

All Q's and Δ's can be determined from the moments of the images. To find the rest of the r's we use the fact that the sum of squares of any row or column in the r matrix is 1.

From the r's we can find directional cosines of the rotation axis and the rotation angle as:

$$\sin\theta = \frac{d}{2} \text{ and } \cos\theta = \frac{d^2 r_1 - (r_8 - r_{6)}^2}{d^2 - (r_8 - r_{6)}^2}$$

where

$$d = \sqrt{(r_8 - r_6)^2 + (r_3 - r_7)^2 + (r_4 - r_2)^2}$$

(both sin θ and cos θ are needed to determine θ uniquely). The direction cosines are:

n1 = (r_8 - r_6) / d
n2 = (r_3 - r_7) / d
n3 = (r_4 - r_2) / d

where $d^2 = (r8-r6)^2 + (r3-r7)^2 + (r4-r2)^2$

We can also find the translation in x and y directions as:

$\Delta x = m_{10} / m_{00}$; $\Delta y = m_{01} / m_{00}$

Δ z is not computable by this method. Also, this method cannot give rotational information for objects which have an axis of reflection symmetry (e.g. parallelograms, triangles) as the tensors all go to zero in such cases. So given an RPP whose attitude has to be determined we need to:

a) Check whether any axis of reflection symmetry exists.

b) Check whether it will have any axis of reflection symmetry under any affine transformation.

c) If the face has any axis of reflection symmetry or will have it under affine transformations, subject it to distortion which removes the axes of reflection symmetry.

The procedures for these steps are as follows:

a) Symmetry conditions for polygons: To check a given polygon for axes of reflection symmetry, we use the concept of the *Voronoi diagram.*

4.2.1) Voronoi diagram: Given a set of N points corresponding to the vertices of the polygon, let x^i and x^j be two of the points. Let P(x^i , x^j) be the half plane containing x^i that is defined by the perpendicular bisector of x^i x^j . The intersection of N - 1 such half planes, denoted by V(i) is called the Voronoi polygon associated with p^i . Note that the polygons are unbounded. For N points there are N such polygons which partition the plane into a net called the *Voronoi diagram.* The construction of the Voronoi diagram for a pentagon is shown in Fig. 2.

Let the vertices of a polygon be $x^1, x^2, ..., x^N$ where

$$x^i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

The points v^{ij} such that

$$v^{ij} = \frac{x^i + x^j}{2}$$

will then be the extremities of the Voronoi diagram.

Given a polygon its Voronoi diagram is constructed and the points v^{ij} obtained. For high complexity polygons it is usually computationally more efficient to use this procedure than to directly evaluate v^{ij} from the x^i and x^j. The symmetry conditions are then determined as explained below.

The symmetry conditions for a polygon depend on whether it has an odd or even number of vertices.

(i) If the number of vertices is odd, the axis of symmetry should pass through a vertex and the mid-point of two other vertices.

(ii) If the number of vertices is even, the axis of symmetry passes through two vertices or two mid-points.

For an odd polygon, an axis of symmetry to exist and pass through a point x^k, the required condition is that there exist a set of x^i and x^j ($i \neq k$, $j \neq k$) such that

$$<x^i - x^j, v^{ij} - x^k> = 0$$

such x^i and x^j will form pairs of points symmetric with respect to the axis through x^k.

If there is to be no axis of symmetry through x^k, then

$$<x^i - x^j, v^{ij} - x^k> \neq 0$$

for $i \neq k$, $j \neq k$. This is a necessary and sufficient condition to guarantee the nonexistence of any axis of symmetry through x^k. The same procedure is repeated for all vertices to verify whether the polygon has any axis of reflection symmetry.

For an even vertex polygon, two different kinds of axes of symmetry can exist.

i) Axis passing through two vertices.

ii) Axis passing through two midpoints of vertices.

i) Let x^a and x^b be the vertices to be checked. An axis of symmetry will pass through these vertices if there exists a set of x^i and x^j such that

$$<x^i - x^j, v^{ij} - x^a> = 0$$

and

$$<x^i - x^j, v^{ij} - x^b> = 0$$

for $i \neq a$ or b, $j \neq a$ or b. Such x^i and x^j points will form pairs which are symmetric with respect to the axis joining x^a and x^b. If the line joining x^a and x^b is not to be an axis of symmetry then

$$<x^i - x^j, v^{ij} - x^a> \neq 0$$

or

$$<x^i - x^j, v^{ij} - x^b> \neq 0$$

for $i \neq a$ or b, $j \neq a$ or b. This is a necessary and sufficient condition to guarantee the nonexistence of any axis of symmetry through x^a and x^b . This procedure is repeated for all vertices to verify whether the polygon has any axis of symmetry.

ii) Let v^{pq} and v^{rs} be the vertex midpoints to be checked. An axis of symmetry will pass through these if there exists a set of x^i and x^j such that

$$\langle x^i - x^j, v^{ij} - v^{pq} \rangle = 0$$

and

$$\langle x^i - x^j, v^{ij} - v^{rs} \rangle = 0$$

for i ≠ p, q, r or s and j ≠p, q, r or s. Such x^i and x^j will form pairs of points symmetric with respect to the axis joining v^{pq} and v^{rs}.

If the line joining v^{pq} and v^{rs} is not to be an axis of symmetry then

$$\langle x^i - x^j, v^{ij} - v^{rq} \rangle \neq 0$$

or

$$\langle x^i - x^j, v^{ij} - v^{rs} \rangle \neq 0$$

for i ≠ p, q, r or s and j ≠ p, q, r or s. This is a necessary and sufficient condition to guarantee the nonexistence of any axis of symmetry through v^{pq} and v^{rs} . This procedure is repeated for all combinations of vertices to check that no axis of symmetry exists.

b) Having verified that no axis of symmetry exists for a polygon, we need to further verify whether any axis of symmetry will exist under any affine transformation of the face.

The conditions derived above for no axis of symmetry to exist are of the general form

$$U^T V \neq 0$$

where U and V are two dimensional vectors. Under an affine transformation A, the condition becomes

$$U^T A^T A V \neq 0$$

whether this condition will be satisfied or not depends on the nature of U and V i.e. on the nature of the polygon and also on what kind of affine transformation it is subjected to i.e. on A. Thus a nonsymmetric triangle can be affine transformed to an equilateral or isosceles triangle which has axis of symmetry.

c) If the face has an axis of symmetry as verified in a) or b) then it is subjected to distortion which removes the axes of symmetry. It has been found that while for any particular distortion there always exists an affine transformation that would yield an axis of symmetry, if the polygon is subjected to two separate distortions which are antisymmetric with respect to each other, there exists no affine transformation which yields axes of symmetry for both cases. If these two distortions are referred to as D_1 and D_2 , then the procedure consists of subjecting the polygon to D_1 and checking it according to a) and b) to see whether it has any axis of symmetry. If it does it is subjected to D_2 and by the above argument it will not have any axis of symmetry.

It has been found that polygons with three or four vertices can always be affine transformed to symmetric polygons. So a *minimum of five points* are needed to obtain a nonsymmetric polygon. A technique has been developed whereby it is not necessary for all the five points to physically lie on the polygon. If we have three points on the polygon, the other two points can be obtained as functions of coordinates of these three points. This is shown for a triangle in Fig.3, where P_1 , P_2 and P_3 are points on the triangle (its vertices) and Q_1 and Q_2 are artificially created points. The five points should be positioned such that the polygon formed by them is the distortion D_1 or D_2 referred to above. Two such distortions are shown in Fig.3 b and c.

4.3) Experimental Results: Fig.4 shows certain objects that have been used to test the simulation of the MIAG algorithm. Fig.5 shows the output of the program for recognition and attitude determination of an object under two different orientations. Fig.6, 7 and 8 refer to certain physical objects that have been used to test the MIAG algorithm. These objects are a simple polyhedral structure, a space shuttle model and a space station model. These figures show these objects and their thinned wireframes.

5) CONCLUSION:

The MIAG algorithm has been extended for the attitude determination of general polyhedral objects. The algorithm has been tested under conditions simulating space conditions and the results are presented in this paper. Work is in progress to extend the algorithm to the general case of recognising and localising any general object.

REFERENCES

[1] C.W.Richard and H.Hemani, "Identification of Three-Dimensional Objects using Fourier Descriptors of the Boundary Curve". *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, Mar.1980, pp127-136

[2] S.A.Dudani, K.F.Breeding, and R.B.McGee, "Aircraft Identification by Moment Invariants", *IEEE Trans. on Computers*, vol C-26, no.1, October 1977, pp. 39-45

[3] L.T.Watson and L.G.Shapiro, "Identification of Space Curves from Two-Dimensional Perspective Views", *IEEE Trans. Pattern Anal. and Machine Intell.* vol. PAMI-4, Sep.1982, pp.469-475

[4] D.Marr and T.Poggio, "Cooperative computation of stereo disparity", *Science*, vol.194, pp.283-287, 1977

[5] D.Marr, Vision: *A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H.Freeman and Co., 1982

[6] O.D.Faugeras, M.Hebert, E.Pauchon, J.Pouce, "Object Representation and Positioning from Range Data," *Robotics Research*, First Intrl. Symp., 1984

[7] B.K.P. Horn "Robot Vision", MIT Press, 1986

[8] K.A. Stevens et al., "Understanding Images at MIT: Representative Progress", *Proc. DARPA Image Understanding Workshop*, 1980, pp 15-19

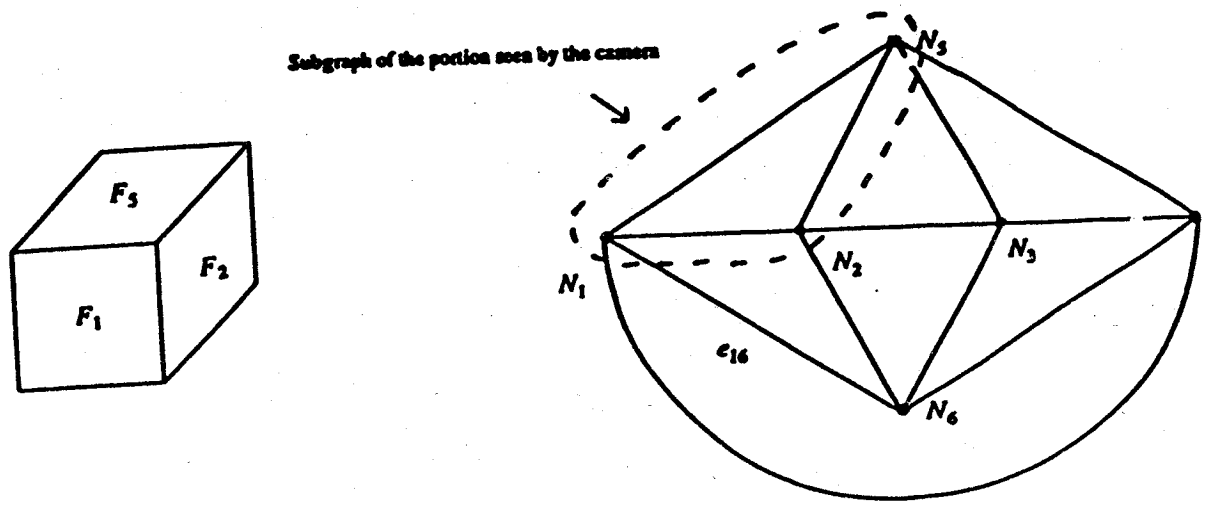
[9] J.R.Kender, "Shape from Texture: A Computational Paradigm", *Proc. DARPA Image Understanding Workshop*, 1979, pp.134-138

[10] A.P.Witkin, "Shape from Contour", Phd dissertation, MIT, Cambridge MA, 1980

[11] H.G.Barrow and J.M.Tenenbaum, "Interpreting line drawings as 3D surfaces" Proc. 1st Annu. Nat. Conf. AI", 1980

[12] Bamieh B. and deFigueiredo R.J.P., "A General Moment Invariants / Attributed-Graph Method for Three-Dimensional Object Recognition from a Single Image", *IEEE Journal of Robotics and Automation*, March 1986, pp 31-41

[13] Pavlidis T., "Algorithms for Graphics and Image Processing", Computer Science Press, 1982.



Object Attributed Graph

Fig.1 Wireframe of a cube and its attributed graph representation

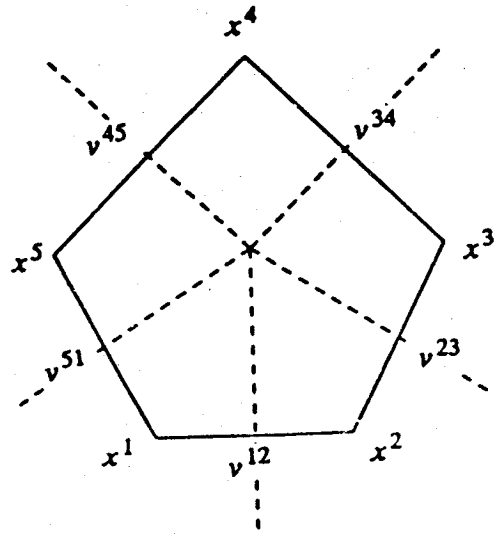
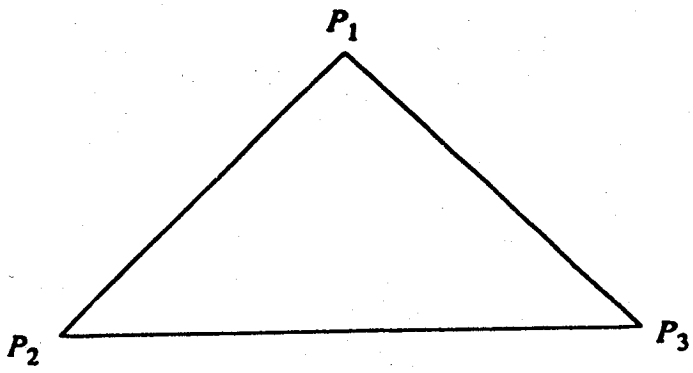
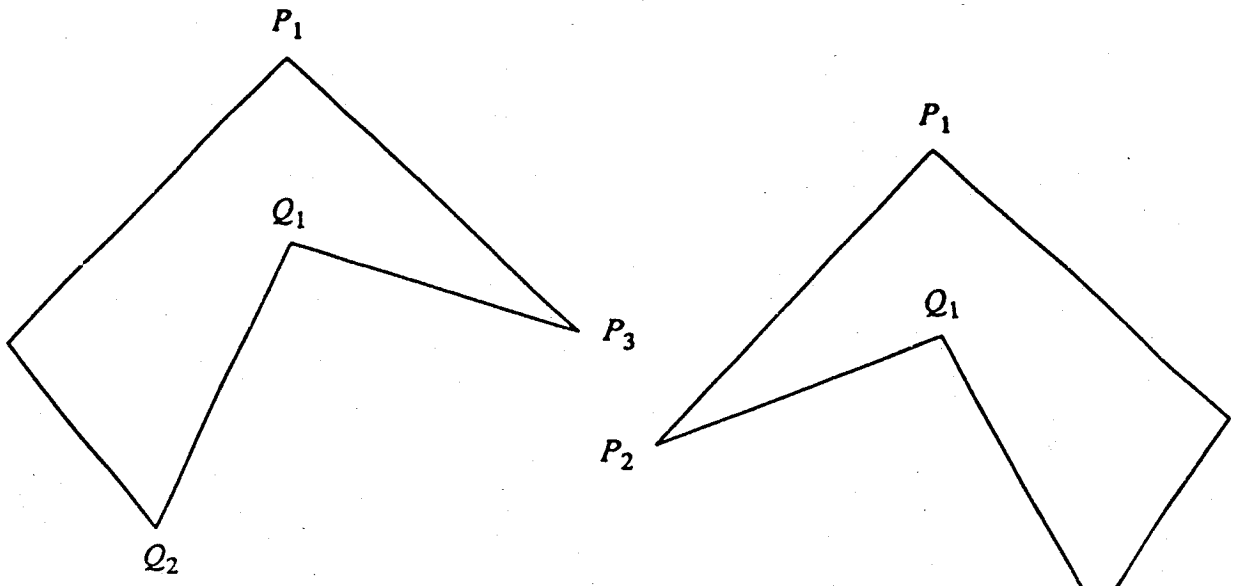


Fig.2 Voronoi diagram of a polygon



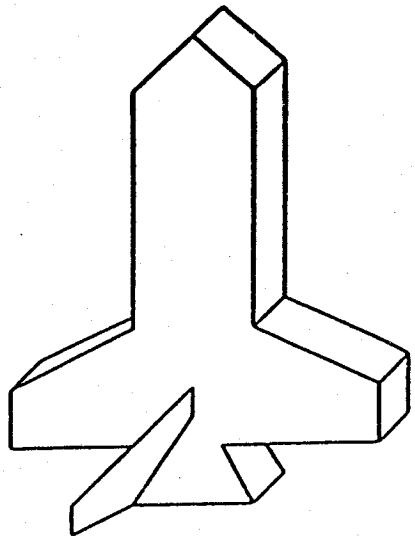
a: Triangle



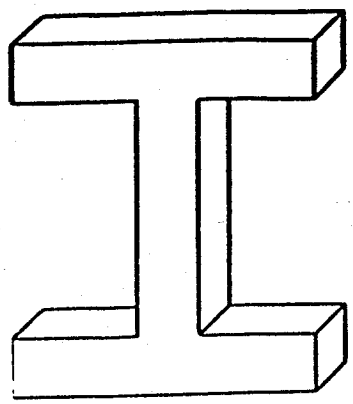
b: D_1

c: D_2

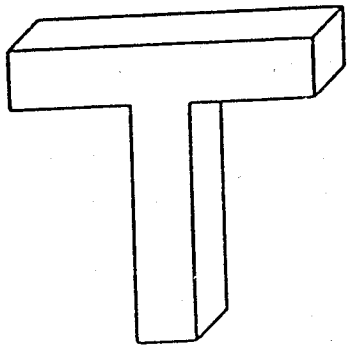
Fig 3: A triangle subjected to D_1 and D_2



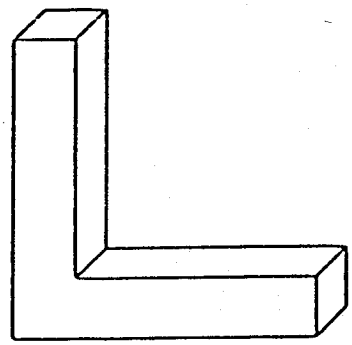
Space Shuttle



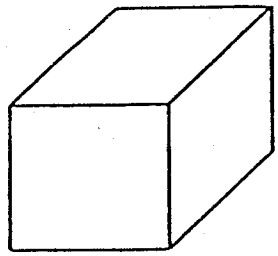
I-Beam



T-Beam

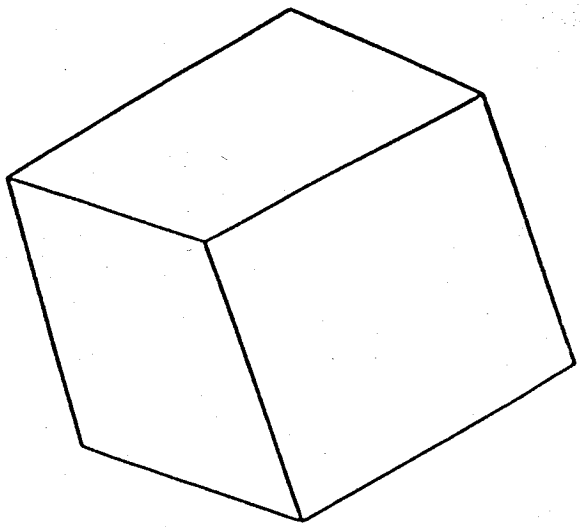


L-Beam



Cube

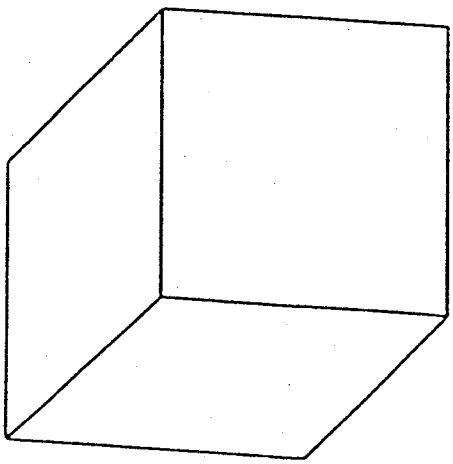
Fig.4 Examples of objects used to test the MIAG algorithm



```
roll = 45
pitch = 45
yaw = 45

Matched to cube

Match correspondences are:
Wireframe face # ---> Model face #
0 ---> 0
1 ---> 1
2 ---> 4
```



```
roll = 90
pitch = 30
yaw = 90

Matched to cube

Match correspondences are:
Wireframe face # ---> Model face #
0 ---> 0
1 ---> 1
2 ---> 5
```

Fig.5 Examples of Recognition and Attitude Determination using MIAG Algorithm

Object ---> Edge Detector Output ---> Edge Thinning Output

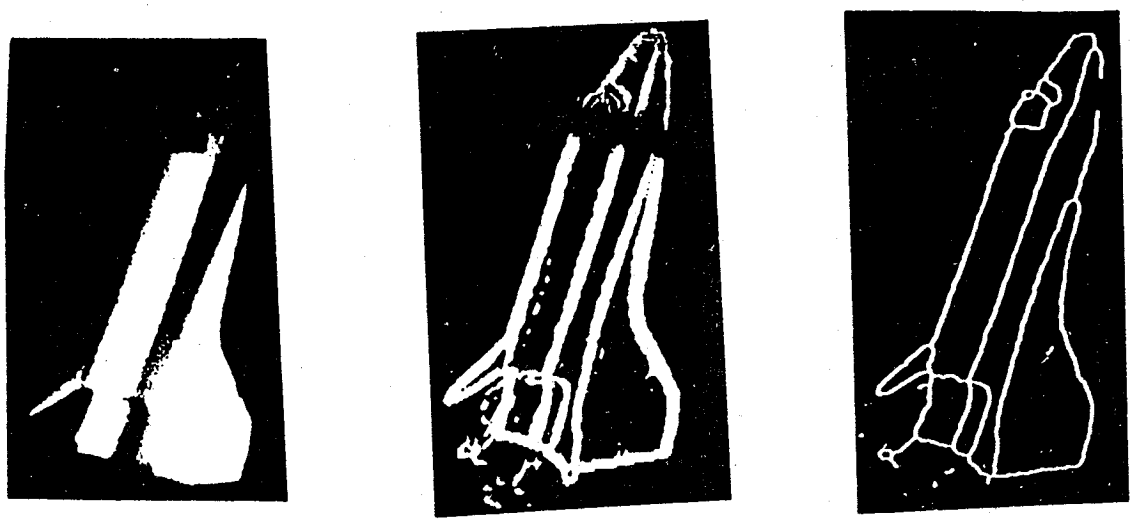


Fig.6 Space Shuttle

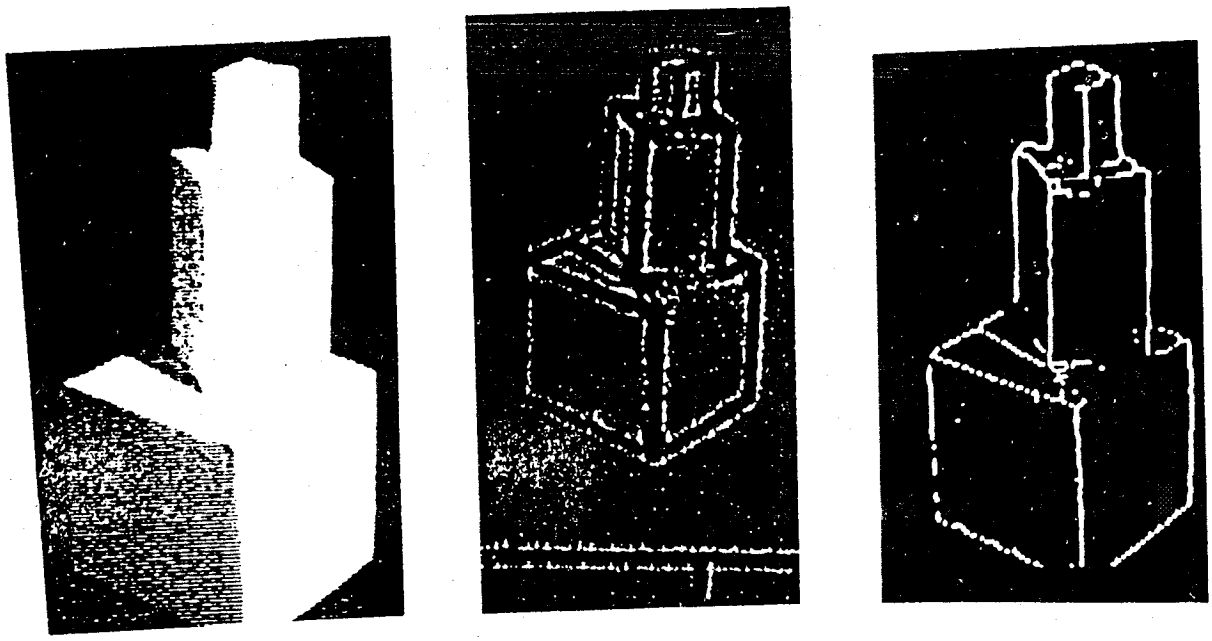


Fig.7 Polyhedral Structure

Object ---> Edge Detector Output ---> Edge Thinning Output

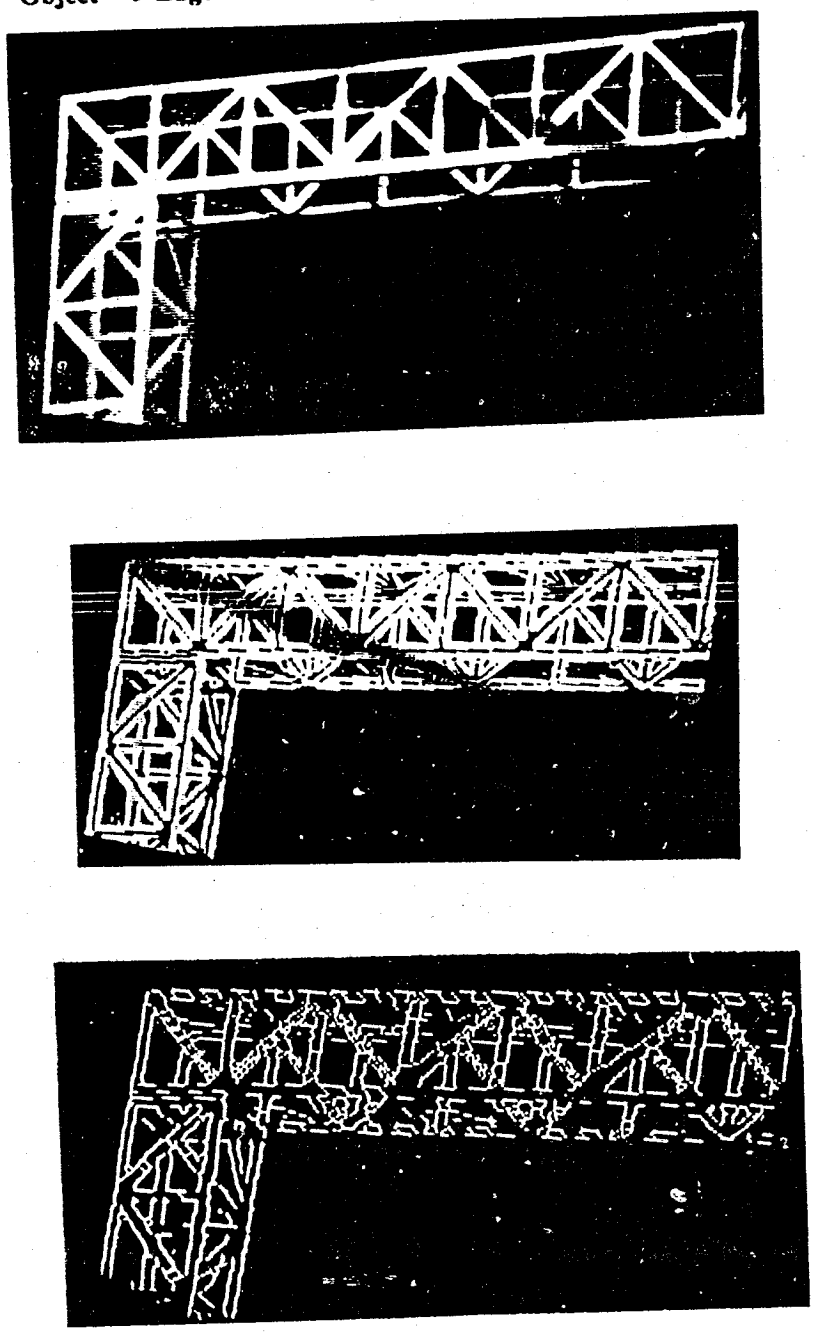


Fig.8 Part of Space Station