# 3-D Object Recognition and Orientation from both Noisy and Occluded 2-D Data

**D.P.Illing*, P.T.Fairney+ & R.J.Wiltshire,***

*Dept. of Mathematics and Computing,
+Dept. of Computer Studies,
The Polytechnic of Wales,
Pontypridd,
Mid Glamorgan,
CF37 1DL

*We present a method of recognising 3-D objects and determining their position and orientation from single 2-D images of both noisy and partially occluded shapes.*

*Our approach combines analytical and structural-type techniques, adopting a 2-D pose-clustering method combined with a comparison to a database of 2-D model descriptions for various object orientations. Object features are used both to control the amount of computation and to direct the search of the database.*

*Features derived from an analysis of the local geometry of the object boundary are used in matching. Local features are preferable to global techniques which perform poorly in noisy conditions and where the object boundary is partially occluded. In these circumstances, local features provide sufficient information to enable the object to be recognised and to determine orientation to within less than seven degrees of arc, on average, in both the major and minor axes of the object.*

## 1. Introduction

Determining 3-D object orientation from single distorted 2-D image frames is a problem in computer vision/pattern recognition.

In our study we have used airborne objects, e.g. aircraft. We have assumed orthographic projection, such that very little, if any, perspective is present in the image, and that only the silhouette projection of the viewed object will be available to us. The method described is not confined to such objects and can be easily adapted to solve for different viewing geometries. However, it is not a general purpose technique because different 3-D objects may project the same 2-D silhouette. In practice this is unlikely to be the case and therefore storing a library of such projections to represent the different orientations of 3-D objects is a feasible approach.

Early work[1,2,3,4,5,6] used global descriptors of shape such as 2-D and 3-D moment invariants and Fourier descriptors; these have been used to recognise or classify 3-D objects from 2-D silhouettes. Given an arbitrary view of a known object, or one of a small set of objects, a set of shape descriptors are computed for the silhouette which are then matched against a library of precomputed descriptors.

These global methods give good results for complete shapes not distorted by noise or occlusion. Moments[7] are particularly susceptible to noise. Neither method is capable of representing local information with any degree of accuracy[8] and both degrade rapidly when part of an object's boundary is missing or occluded. As a result there has been a move towards local descriptions of shape, i.e. corners[9,10], holes[10], lines [11] and curvature[12].

The generalised Hough transform[13] is commonly used [11,14,15,16,17]. Such approaches involve an accumulation of evidence which either supports or weakens a hypothesis on the pose of an object in the image. Objects are recognised by means of a rotation, scaling, translation (RST) transformation which maps observed image features onto features in the model. This is achieved by assuming that the object in the image can be represented by an instance of the model having undergone some RST transformation and that the object is rigid. Even with such limitations the approach is viable. A match is still possible even if some model features are missing, and if spurious features are found. As a result the method is able to cope with both noisy and partially occluded objects.

Only the object's 2-D silhouette is available and we have therefore chosen boundary segments as our local features. These have been used previously to register aerial imagery with maps, to recognise instances of objects in aerial photographs and to match 2-D models to images of 2-D industrial parts[11].

We use wire frame models of a Hawk jet trainer and a missile to generate both model descriptions and test data for our system. Analytic techniques are used to derive structural descriptions of object boundaries, in terms of angles and straight line segments. These features are then used in a pose-clustering process. When this is combined with matching to a database of precomputed model descriptions we are able to recognise and determine the unconstrained orientation of a 3-D object from a single image of its 2-D silhouette projection with an accuracy of less than seven degrees, on average. The position of the object within the field of view is unimportant, as is the scale of the object with respect to its model. Given the focal length of the camera then the resulting scale factor can be used to estimate object range.

We use object features to limit the combinatorics of the clustering and to reduce the *im* combinations of *m* model segments and *i* image segments by appproximately 90%. By using simple measures of shape to direct the search of the database, less than 20% of the model database entries are selected as possible poses for our matching algorithm. Heuristics limit the number of hypothesised poses that are verified to about 30% of the selected model database entries, and of these only 15% will complete this final stage.

## 2. The Pose-clustering Paradigm

Consider figure 1 which shows a single image vector $\vec{i}$ and a single model vector $\vec{m}$. The RST transformation which will bring $\vec{i}$ into registration with $\vec{m}$ may be computed as follows.



BMVC 1990 doi:10.5244/C.4.38

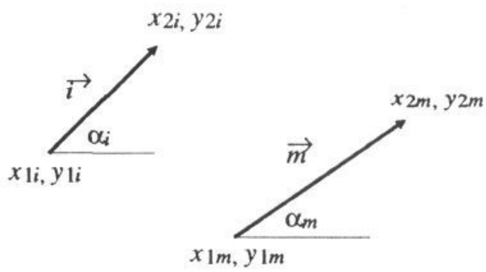


***Figure 1. An RST transformation.***

The rotation angle, $\alpha$, is given by

$$\alpha = \alpha_m - \alpha_i \qquad (1)$$

where $\alpha_m$ is the orientation of the model vector, $\vec{m}$, and $\alpha_i$ is the orientation of the image vector, $\vec{i}$. The scale change, $\rho$, is given by

$$\rho = \frac{\lambda_m}{\lambda_i} \qquad (2)$$

where $\lambda_m$ = length of model vector, $\vec{m}$

and $\lambda_i$ = length of image vector, $\vec{i}$.

The translations, $\Delta x$ and $\Delta y$ are then given by

$$\Delta x = \rho\, y_{1i} \sin\alpha - \rho\, x_{1i} \cos\alpha + x_{1m} \qquad (3)$$

$$\Delta y = -\rho\, x_{1i} \sin\alpha - \rho\, y_{1i} \cos\alpha + y_{1m}. \qquad (4)$$

The resulting transformation, in homogeneous coordinates, is then

$$[\, x_m, y_m, 1\,] = [\, x_i, y_i, 1\,] \begin{bmatrix} \rho\cos\alpha & \rho\sin\alpha & 0 \\ -\rho\sin\alpha & \rho\cos\alpha & 0 \\ \Delta x & \Delta y & 1 \end{bmatrix}. \qquad (5)$$

Consider an image which comprises a series of points, each of which represent a feature of the boundary of the object under scrutiny e.g. points of maximum curvature. These points are connected by straight line segments such that our image now comprises a series of directed line segments or edge vectors. Next, consider a similar image which will be our model. To bring the test image into registration with the model, we must find the RST transformation which optimally brings the object edge vectors into alignment with the corresponding model edge vectors.

We use a pose-clustering procedure to achieve this. Consider the image vector $\vec{a_i}$ in figure 2, we compute an RST transformation which will bring this into registration with the model vector $\vec{a_m}$. This transformation comprises a scale change, $\rho$, a rotation angle, $\alpha$, and two translations, $\Delta x$ and $\Delta y$. The computed transformation can be considered to contribute a count of one to the particular cell, or bin, in the 4-D space indexed by $\rho$, $\alpha$, $\Delta x$ and $\Delta y$. We can also compute an RST transformation which will bring this vector into alignment with all of the other model vectors $\vec{b_m}, \vec{c_m}$. These transformations also contribute counts to their respective bins in the 4-D space. Next image vector $\vec{b_i}$ is considered and RST transformations are computed which register it with all of the model vectors. At the same time the counts associated with the corresponding bins of the 4-D space are incremented. This procedure is followed until all image vectors have been considered. The correct transformation will show up as the most significant cluster in this 4-D space. The process of assigning data quads, representing the transformation parameters, to cells in this 4-D space we term binning.

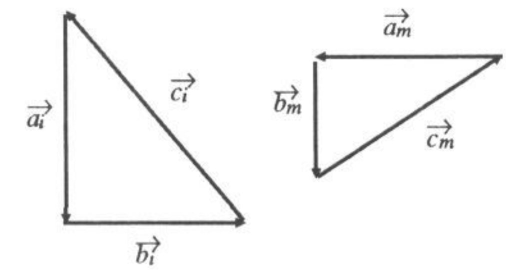


***Figure 2. Both image and model data must be brought into registration.***

We use a hierarchical approach and a series of overlapping bins in order to prevent the splitting of clusters. Several cluster centres may be followed up at any given level, and in the final level the best cluster is chosen from several candidates by computing "the average degree of alignment " between the image and model edge vectors. The cluster with the smallest r.m.s. distance between the vectors provides the final match which is also optimum in a global sense.

## 3. Determining 3-D Orientation

Rather than just determining localisation and orientation in one plane, (i.e the plane of the image) we are able to determine the orientation of an object in 3-D space and its position within the field of view.

Orientation is given by three rotation angles, one about each axis of our coordinate system. The terms yaw, roll and pitch are commonly applied to airborne objects.The rotation about the z-axis (pitch) is a rotation in the plane of the image. The rotation about the y-axis (yaw) is a rotation in and out of the plane of the image and the rotation about the x-axis (roll) is a rotation about the longitudinal axis of the object itself.

For any fixed yaw and roll angles the problem reduces to 2-D. Envisage a database containing descriptions of a model object in terms of its edge segments for a series of yaw and roll angles corresponding to different 2-D projections, then, given an image of this object at some unknown orientation, it is possible to register this image with model elements in the database. Some combinations of yaw and roll give poor registration but for orientations close to the true orientation of the object the registration will be good. The problem now reduces to that of finding the optimum registration of the model edge vectors in the database with those of the object. A hypothesise and test type of approach is used in which simple measures of shape are used to direct a search of the database. Each selected model entry represents a hypothesised pose of the object in our image. A pose-clustering procedure is used to derive the parameters of the RST transformation which will bring the object into registration with its model description. The validity of the hypothesised pose is tested by applying the resulting transformation to the image data and comparing the r.m.s distance between the transformed image and model edge vectors. The database entry with the smallest r.m.s distance provides the final match. The yaw and roll angles are given by the yaw and roll angles of the selected entry in the database while the pitch angle is produced by the registration process.

## 4. Equipment Used

We use a PC compatible fitted with an Imaging Technology Inc. FG-100-AT board. This provides a single, 12 bit image plane of 1024 * 1024 pixels. A single image occupies 512 * 512 pixels.

Unused portions of the image plane are used to store data in the form of integer values in the range of 12 bits, i.e. 0-4095. We store our database on an unused portion of the image plane.

## 5. Feature Extraction

Man-made objects such as aircraft, missiles etc. have well defined geometric outlines. The silhouette of such shapes can therefore be considered as a series of straight line segments separated by small regions where the curvature may change rapidly, perhaps discontinuously. One approach is therefore to locate points of high curvature and connect them by straight lines. This method has considerable support from psychological experiments[18] and as a means of shape description has become increasingly popular[9,12,19,20].

Our experiments indicate that, when noise is present, the performance of such algorithms is unreliable and therefore, locating high curvature points directly is ineffective. Instead, our approach is to detect the constituent sides of the object boundary. We search the boundary curve for linear segments and at the junctions between boundary segments we assign curvature discontinuities. The use of estimates of general trends in the curves to derive point features, rather than attempting to locate the features themselves, means that the detected positions of rapid curvature changes are much more stable in the presence of noise which is effectively filtered out.

We fit straight line segments of increasing length to the boundary curve using a least-squares minimisation. This is done for a series of points around the boundary which are equidistant in arc length. Then for each line length we determine points around the boundary where the errors of fit are locally minimal producing a sub-set of the original fitted lines. These points should correspond to regions of the boundary which approximate straight lines. The resulting lines are hierarchically assigned to an initially empty map of our object boundary, such that the longest lines are assigned first. When a line is assigned its domain on the boundary map is marked as bound; no other lines can subsequently be assigned within this domain. This process is continued until either there are no more lines to fit, or the map becomes full. We extrapolate assigned lines across any small gaps which might remain in the map and determine their intersection points. These points become the object features, corresponding to the points of maximum curvature.

## 6. The Model Data

We generate a database of model edge arrays using wire frame models of a missile and a Hawk jet trainer. Each wire frame model is rotated through increments of five degrees in yaw & roll, using a standard transformation matrix[21]. At each orientation a description of the model in terms of line segments is produced. The resulting list of x,y coordinate pairs constitutes the model description at that orientation. We will refer to these model descriptions, which go to make up our database, as model arrays. The description of the object in the image we will term the image array. The model arrays are stored on a spare quadrant of the image plane.

## 7. Retrieving Model data

When searching the database for a best match we wish to limit the number of model arrays we read and compare, to a small but suitable subset of the database. With this aim, we devised a method of pointers to individual model arrays based upon the area/perimeter$^2$ ratio, $r_m$. We compute the area/perimeter$^2$ ratio of the viewed object, $r_i$, and this is used to direct our search of the database. We scan those model arrays whose ratio, $r_m$, is within the range

$$r_i - t \leq r_m \leq r_i + t \qquad (6)$$

where $t$ is a predetermined threshold.

The values $r_i \pm t$ represent the upper and lower bounds of the area/perimeter$^2$ ratio of model data we wish to compare. Having located candidate model arrays in this way we match each with the image array; this match is described next.

In practice, this limits searches to less than 20% of the database.

## 8. The Registration Process

The feature detection algorithm produces a list of x,y coordinates which correspond to end-points of linear segments around the boundary of the object. Each vertex connecting adjacent segments is assigned a type depending upon the angle between the segments. For angles in the range $[0, \pi]$, a type "convex" is assigned, whilst for angles in the range $[\pi, 2\pi]$, we assign a type "concave".

The area/perimeter$^2$ ratio of the description is computed and model database records whose area/perimeter$^2$ ratio lies within the specified range of this value are selected for comparison. Then, the object description is compared to each selected model record in turn through a pose-clustering procedure.

For each selected record only combinations of image and model segments with the same vertex types are included when forming the 4-D cluster space. Also, combinations of model and image edge vectors are rejected if the computed rotation angle does not fall within ±15 degrees of the difference in the principal axes of the object and model. This is a robust procedure to adopt since, even in the case of corrupt data, the principal axis should not be in error by more than this amount. This use of object features reduces the number of possible combinations of image and model arrays by about 90%. Further reductions in the number of hypothesised poses are achieved by using the difference in the coordinates of the centroids of the object and model to constrain the translations $\Delta x, \Delta y$. We also use the ratio of the square root of the projected area of object and model to constrain scale factors.

Initially the cluster space will consist of many different combinations of model and image edge vectors. It is not possible for any one image edge vector to correspond to more then one model edge vector and vice versa. The number of unique model-object edge vector pairs in our cluster space is computed and if below a predetermined threshold then the clustering process is terminated. This threshold is set at 40% of the number of model edge vectors.

The algorithm used to identify candidate clusters or transformations lies at the heart of the method and therefore needs to be carried out efficiently. An approach which is hierarchical in nature is described, and rather than clustering in 4-space directly and forming a 4-D histogram with its associated high storage requirements, a simple 2-D table structure is used.

The parameters of each computed candidate transformation are examined in turn. This information is entered into an initially empty table which comprises five columns $\rho$, $\alpha$, $\Delta x$, $\Delta y$ and a count. If a given combination of parameters is not present then that combination is entered and the count initialised to one. If already present then the count is simply incremented. In practice, a single record in the table will not represent a single transformation, but rather, is a bin containing a group of transformations whose parameters lie within a specified range of each other.

This procedure is performed in a hierarchical fashion, binning up the data initially on a coarse scale, locating the candidate clusters at that level which are to be further binned in subsequent levels. We set a maximum number of such iterations to be five and terminate the binning when this is exceeded or when the standard deviation within individual bins falls within acceptable bounds. In practice, the process converges after only two or three iterations, and cases which continue beyond this point are probably mismatches with our database. In this instance there will be very few possible model-object edge vector pairs in the final bin. To prevent clusters from splitting into different bins, we use a series of overlapping bins.

The data in the finally selected bin gives an estimate of the RST transformation which will register our object data with the database entry. We now wish to evaluate how good this registration is - poor registration indicates an incorrect match. The estimated transformation is evaluated by applying the transformation itself to the x,y coordinates which correspond to the data in the final bin. We then compute the r.m.s difference between these transformed object coordinates and those of the model; this is used to select the best match. The smallest r.m.s error, of those model arrays selected, produces the estimated orientation of our sensed object. Using object features means that less than one third of the selected database entries reach the stage of hypothesis verification. To prevent complete computation of the poorest matches, we set an initial threshold on the r.m.s. value, and once this value is exceeded the summing and squaring operations cease. When a complete pass through a data set is accomplished a new minimum r.m.s. value is obtained and this becomes the new threshold value. We find that as few as 5% of the selected database entries reach this final stage of the matching process.

When the model data for several different objects are combined to produce a single multi-object database, the system is also able to perform object recognition, as our results show. In this case, the best match also determines the objects identity.

## 9. Experimental Results

Wire frame models of a Hawk jet trainer and a missile were used to generate object data to test our system. The wire frame models were rotated into many orientations and object descriptions generated, (section 5). Each computed description was then matched with our database. This was performed for both clean images and noisy images. We added noise[22], in varying amounts, to the traced outline of each wire frame model and then used these noisy boundaries to generate line segment descriptions of our objects for matching to our database. The results in Table 1 are for a noise level which perturbed ≈60% of boundary pixels. Figure 3, shows a typical noisy object with ≈60% of the boundary pixels perturbed. We also tested our system with partially occluded objects. Partially occluded boundaries were generated by producing solid silhouettes from the wire frame models and then masking off selected regions. The boundaries of the

| orientation | | | | | | rec. | obj. | error in degrees of arc | |
|---|---|---|---|---|---|---|---|---|---|
| true | | | sensed | | | | | | |
| yaw | roll | pitch | yaw | roll | pitch | | | $\theta$ | $\phi$ |
| 3 | 12 | 11 | 5 | 5 | 10.4 | yes | m | 2.2 | 7.0 |
| 27 | 36 | 5 | 25 | 45 | 7.1 | yes | m | 2.7 | 8.5 |
| 61 | 37 | 123 | 60 | 30 | 121.4 | yes | m | 1.4 | 5.3 |
| 82 | 39 | 359 | 80 | 35 | 4.4 | yes | m | 2.1 | 9.1 |
| 79 | 81 | 315 | 75 | 80 | 317.8 | yes | m | 4.1 | 5.6 |
| 3 | 12 | 11 | 5 | 15 | 12.5 | yes | h | 2.8 | 3.2 |
| 27 | 36 | 5 | 25 | 35 | 4.0 | yes | h | 2.2 | 0.7 |
| 61 | 37 | 123 | 60 | 40 | 123.2 | yes | h | 1.0 | 3.1 |
| 82 | 39 | 359 | 80 | 30 | 5.4 | yes | h | 2.2 | 15.1 |
| 79 | 87 | 315 | 80 | 80 | 317.4 | yes | h | 1.1 | 3.1 |

***Table 1. Results for noisy images of both the Hawk(h) and the missile(m) with ≈60% of the boundary pixels perturbed. (rec = recognised; obj = object).***

***Figure 3. A noisy( ≈60%) image of the Hawk jet trainer.***

| orientation | | | | | | rec. | obj. | error in degrees of arc | |
|---|---|---|---|---|---|---|---|---|---|
| true | | | sensed | | | | | | |
| yaw | roll | pitch | yaw | roll | pitch | | | $\theta$ | $\phi$ |
| 45 | 11 | 39 | 50 | 10 | 35.8 | yes | m | 5.4 | 3.1 |
| 82 | 39 | 359 | 85 | 40 | 356 | yes | m | 3 | 4.5 |
| 45 | 11 | 39 | 50 | 20 | 42.9 | yes | h | 5.7 | 6.2 |
| 82 | 39 | 359 | 85 | 10 | 328 | yes | h | 4.5 | 4.7 |

***Table 2. These results are for noisy images with 40% of the boundary removed. Both the complete and partial object is shown alongside each result. The applied noise level resulted in ≈60% of boundary pixels being perturbed.(h=hawk; m=missile; rec=recognised)***

resulting clipped shapes were then traced and our line fitting algorithm used to generate the object descriptions. Table 2, shows a selection of our results, in each case both the complete and the occluded silhouette is shown. When noise levels reached ≈60% and the amount of occlusion applied was 40%, the system began to break down, i.e. approximately 1 in 6 test objects were unrecognised.

Table 3. shows the average results for all test images.

Because some orientations are not unique, i.e. different combinations of yaw, roll and pitch produce identical orientations, some apparently disparate sensed orientations can in fact be quite accurate. Therefore, in order to quantify our results we define errors in terms of two angles $\theta$, $\phi$, which are the difference in degrees of arc in the major and minor axes of both sensed and true orientations respectively.

| Proportion of occluded boundary. | Applied noise level | | | |
|---|---|---|---|---|
| | ≈30% | | ≈60% | |
| | uncertainty in degrees of arc | | | |
| | $\theta$ | $\phi$ | $\theta$ | $\phi$ |
| 0 | 4.4 | 4.5 | 4.8 | 4.5 |
| 10% | 5.8 | 3.6 | 3.0 | 3.8 |
| 20% | 6.4 | 3.7 | 6.5 | 5.9 |
| 30% | 4.4 | 6.6 | 6.1 | 4.8 |

***Table 3. Average results for all images***

For objects with a high degree of symmetry, i.e. the missile, some different orientations may produce the same silhouette. As an example, our method will not distinguish between the orientations 0,0,0 and 0,90,0.. However, the human eye could not distinguish between these orientations without the aid of other cues. As a result, we only store and therefore detect roll angles for the missile in the range [0....85] degrees;

## 10. Discussion

Noise and occlusion will distort an object's shape sufficiently to conceal, or even add, features. Local methods, which match parts of a boundary, are more likely to succeed under such conditions than global shape descriptors.

Using a least squares fit to derive a series of linear segments which represent the local geometry of the boundary of an object is effective in reducing the influence of noise and is a robust method of describing man-made shapes. The algorithm is also capable of describing small changes in the orientation of modelled objects.

The combination of a clustering algorithm to derive the parameters of the RST transformation which maps image data onto model data with a search of a database of 2-D model descriptions forms the basis of a system which is capable of determining object orientation to within an average uncertainty of 7 degrees of arc in both the major and minor axes, even when a high degree of noise is present. This method also produces accurate estimates of the object's position, and its scaling with respect to its model description.

Because the method matches individual image edge segments to model edge segments it is able to cope with occluded or partial data. We have found that the method consistently gives good results with up to 30% of the boundary missing and with noise affecting $\approx 60$ % of the remaining edges. When 40% of the boundary is removed the method occasionally fails to recognise the object and therefore determine orientation. Nevertheless we have shown that the method is reasonably robust to both noise and to occluded objects.

Precomputing model data and storing it on an unused quadrant of the image plane is both effective and efficient, since the database is effectively held in RAM. Furthermore, the model descriptions are computed once only in a process that occurs prior to the primary functions of the system. The use of object features and heuristics which reduce the search area and the computational demands of subsequent matching further enhances the method.

We have analysed how errors in the detected positions of the object features affect the calculated RST transformation. Assuming that the model feature positions are known exactly, the errors in the transformation parameters will be due only to errors in the coordinates derived in the image frame.

Error analysis is beyond the scope of this paper. However, the practical implications are as follows. We wish to use the longest line lengths available to us in order that we minimise errors. Simply deleting short lines is inadvisable since, by doing so, we will be removing some features which are 'close' to one another from consideration by our clustering algorithm; these may be important in deriving orientation.

The procedure we have adopted is to form a series of virtual edges, that is edges which connect features that are not directly linked by real edges in the image. In this way we can include every detected feature, whilst choosing our virtual edges such that the edge lengths are maximised. In this way we minimise errors in the parameters of the resulting RST transformation, and thus improve the accuracy of the determined orientation.

## 11. Conclusions

Redefining boundaries by directed edge vectors and matching them to similarly described models provides a sound basis for recognising objects and determining their orientation to within an acceptable degree of uncertainty. When a pose-clustering technique is combined with a search of a database of precomputed 2-D descriptions, the result is an effective and accurate algorithm.

## References


1. **Dudani, S.A., Breeding, K.J .and McGhee, R.B**. 1977 *"Aircraft Identification by Moment Invariants"*, I.E.E.E. Trans. Computers, C-26, no.1, pp39-46.

2. **Reeves, A.P. , Prokop, R.J., Andrews, S.E. and Kuhl, F.P**. 1984 *"Three-dimensional Shape Analysis Using Moments and Fourier Descriptors"*, Proc. 7th Int. Conf. Pattern Recogniton, Montreal, Canada, July 30 - Aug. 2.

3. **Zahn, C.T. and Roskies R.Z.** 1972 *"Fourier Descriptors for Plane Closed Curves"*, I.E.E.E. Trans. Computers, C-21, pp 269-281.

4. **Granlund, G.H.** 1972 *"Fourier Preprocessing for Hand Print Character Recognition"* I.E.E.E. Trans. Computers, C-21, pp195-201.

5. **Richards, C.W. and Hemami, H.** 1974. *"Identification of Three-dimensional Objects using Fourier Descriptors of the Boundary Curve"*, I.E.E.E. Trans. Systems, Man, Cybern. SMC-4, pp371- 378.

6. **Wallace, T.P. and Wintz, P.A**. 1980 *"An Efficient Three-dimensional Aircraft Recognition Algorithm Using Normalised Fourier Descriptors"*, Comput. Graphics and Image Processing, vol 13, pp 96-126.

7. **Teh, C.H. and Chin R.T**. 1988 *"On Image Analysis By The Method Of Moments"* I.E.E.E. Trans. Pattern Analysis Mach. Intelligence, PAMI-10, no.4, pp496-513.

8. **Rice, J,R,** 1969 *"The Approximation of Functions,2 "*, Addison-Wesley,Reading.

9. **Koch, M.W. and Kashyap, L.** 1987 *"Using Polygons To Recognise And Locate Partially Occluded Objects"* I.E.E.E. Trans. Pattern Analysis Mach. Intelligence, PAMI-9, no 4, pp483- 494.

10. **Bolles, R.C. and Cain R.A**. 1982 *"Recognising And Locating Partially Visible Objects : The Local-Feature-Focus Method"* Int. Journal Robotics Res., vol 1, pt.3,pp57-82.

11. **Stockman, G., Kopstein, S., Benett, S.** 1982 *"Matching Images to Models for Registration and Object Detection via Clustering."* I.E.E.E. Trans Pattern Analysis Mach. Intelligence, PAMI-4, no.3, pp229-241.

12. **Wallace, T.P., Mitchell, O.R. and Fukunaga, K.** 1981 *"Three-dimensional Shape Analysis Using Local Shape Descriptors"*, I.E.E.E. Trans. Pattern Recognition Mach. Intelligence, PAMI-3, no.3, pp310-323.

13. **Ballard, D.H.** 1981 *"Generalising The Hough Transform To Detect Arbitrary Shapes"*, Pattern Recognition, vol.13, no.2, pp111-122.

14. **Dhome, M. and Kasvand,T**. 1987 *"Polyhedra Recognition by Hypothesis Accumulation"*, I.E.E.E. Trans. Pattern Recognition Mach. Intelligence, PAMI-9, no.3, pp429-438.

15. **Turney, J.L., Mudge, T.N. and Volz, R.A.** 1985 *"Recognising Partially Occluded Parts"*, I.E.E.E. Trans Pattern Recognition Mach. Intelligence, PAMI-7, no.4, pp410-421.

16. **Stockman, G.** 1987 *"Object Recognition And Localisation via Pose-clustering"*, Computer Vision, Graphics and Image Processing, vol. 40 pp 361-387.

17. **Stockman, G. and Esteva, J.C.** 1985 *"3-d Object Pose From Clustering With Multiple Views"*, Pattern Recognition Letters, vol.3, pp279-286.

18. **Attneave, F.** 1954 *"Some Informational Aspects Of Visual Perception"* Psychological Review 61, pp183-193.

19. **Shirai, Y.** 1978 *"Recognition Of Real-world Objects Using Edge Cues"*, in Computer Vision Systems (A Hanson, E.M. Riseman, Eds.), Academic Press, New York. pp 353-362.

20. **Chien, C.H. and Aggarwal, J.K.** 1989 *"Model Construction And Shape Recognition From Occluding Contours"*, I.E.E.E. Trans. Pattern Analysis Mach. Intelligence, PAMI-11, no.4, pp372-389.

21. **Foley,J.D. & Van Dam,A**. *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1983.

22. Theoretical Weapon Attitude Study, Report No.92.53/4, M.O.D(PE), R.A.E., Jan.1986.