Journal of Intelligent and Robotic Systems **18:** 351–365, 1997.

A CAD Model Based System for Object Recognition

JHARNA MAJUMDAR*
Aeronautical Development Establishment, Defence Research and Development Organisation (DRDO), C.V. Raman Nagar, Bangalore 560 093, India; e-mail: jharna@ade.ernet.in

A. G. SEETHALAKSHMY
Centre for Artificial Intelligence and Robotics, Defence Research and Development Organisation (DRDO), Raj Bhavan Circle, High Grounds, Bangalore 560 001, India.

(Received: 9 August 1995; accepted 30 September 1996)

Abstract. 3D object recognition is a difficult and yet an important problem in computer vision. A 3D object recognition system has two major components, namely: an object modeller and a system that performs the matching of stored representations to those derived from the sensed image. The performance of systems wherein the construction of object models is done by training from one or more images of the objects, has not been very satisfactory. Although objects used in a robotic workcell or in assembly processes have been designed using a CAD system, the vision systems used for recognition of these objects are independent of the CAD database. This paper proposes a scheme for interfacing the CAD database of objects and the computer vision processes used for recognising these objects. CAD models of objects are processed to generate vision oriented features that appear in the different views of the object and the same features are extracted from images of the object to identify the object and its pose.

Key words: model-based computer vision, perspective view, vanishing point, relational graph structure

1. Introduction

Computer vision systems are gaining wider acceptance in industries for a variety of applications. The first step in such a system is almost always the object recognition. The development of such 3D object recognition systems have been occupying the attention of researchers for years. Computer vision can be used in the industry in a more effective and flexible way if a closer tie is established between the computer vision process and the CAD database. A systematic approach is therefore required, where the CAD model may be used to generate the object representations as well as the recognition strategies. In such a system, the CAD models of the objects and the knowledge of the working environment provides the basis for driving the vision system. A set of vision oriented features are extracted by post processing the CAD model data and the same features are extracted from images of the objects. Since both the model and the image are

* Author for correspondence.

described in terms of the same features, image to model correspondence can easily be established to identify the object and its pose.

2. Literature Survey

Model based computer vision has been an active area of research in recent years. A survey of model based object recognition systems is reported in [1]. Both vision and CAD systems rely on models of objects involved, but these two fields have always remained disjoint, possibly due to their different modes of data representation. The various strategies for data representation in the two worlds, namely CAD and image processing, are discussed in [2]. Several research results where conventional CAD tools are used for 3D object recognition are reported in [3–5].

Baumgart [6] developed a 3D geometric modelling system (GEOMED) for computer vision applications. Hermen [7] established the three dimensional structure of the visible faces of an object from a single view by assuming the object to have the shape of a polyhedron with adjacent perpendicular faces and edges. Koshikawa et al. [8] have used a solid modeller (GEOMAP) to find the stable positions of an object from the observed surface normals. Bolles et al. [9] have designed a CAD model based vision system (3DPO) that recognizes partially hidden objects, determines their location and orientation, and decides the order in which a robot should pick each part. Henderson et al. [10] have used a Computer Aided Geometric Design (CAGD) system for visual recognition and manipulation of objects. Recent work by Ho [11, 12] has focused on the generation of computer vision models directly from a CAGD model.

More recently, A. K. Jain et al. [13] have developed a system which uses 3D object descriptions created on a CAD system to generate both view-dependent and view-independent relational graphs for use in object recognition. R. Jain et al. [14] have reported a system that uses the information available in a CAD database for efficient feature selection for object recognition.

CAD based object recognition systems have mainly been used for visual inspection applications. An intelligent inspection planning subsystem for a CIM unit manufacturing mechanical parts has been discussed in [15]. Park and Mitchell [16] have established a framework for automated visual inspection using CAD database of parts.

3. Proposed Scheme

In this paper we propose a scheme to recognize 3D planar objects and to identify their poses. The objects are assumed to lie on a worktable in one of their stable poses. The viewing direction for the visual sensors is along the Z direction, which is perpendicular to the worktable plane. The object recognition system works in two modes, namely the off-line mode and the run-time mode. In the

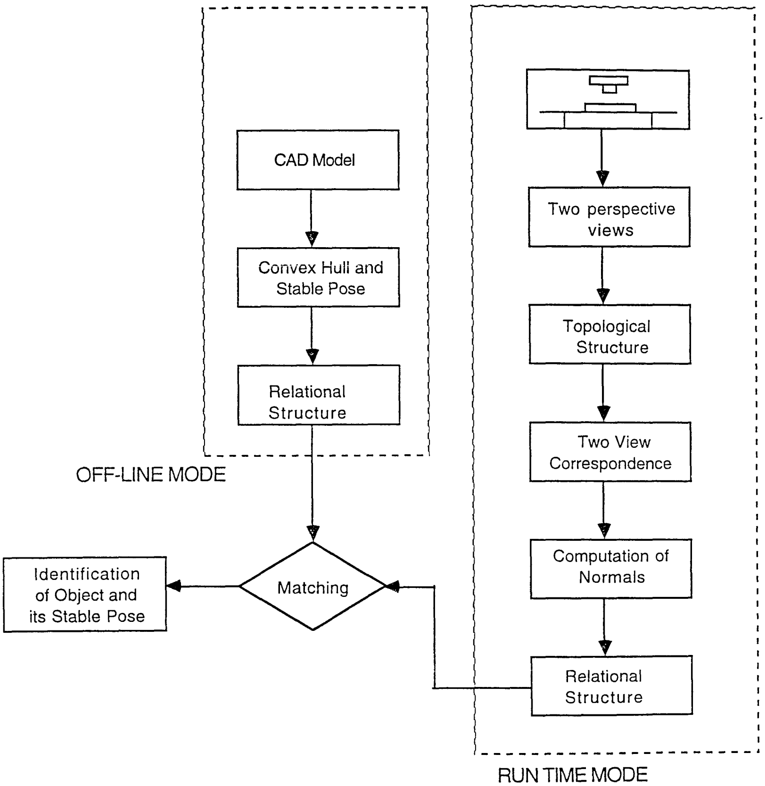


Figure 1. Flow diagram of the proposed scheme.

off-line mode, vision oriented data derived from a 3D CAD system database are used to generate the model database for object recognition. The CAD model of the parts has been used to obtain a relational graph representation of the parts in various stable poses, which is then used for their recognition. Various procedures have been applied to the CAD system output to obtain vision oriented features that appear in the different views of the object. These features are invariant as to location, orientation and scale of the object. In the run-time mode, two perspective views of an object in one of its stable poses are taken, and analysed to extract the same set of invariant features, which are consequently matched to identify the object and its pose. The flow diagram of the proposed scheme is shown in Figure 1. The system consists of the following three modules namely:

- Construction of model database;
- Extraction of image features;
- Matching model data with image data.

3.1. CONSTRUCTION OF MODEL DATABASE

In the proposed work, object models have been constructed directly from their CAD data without the use of images. This process consists of the four steps sketched out below.

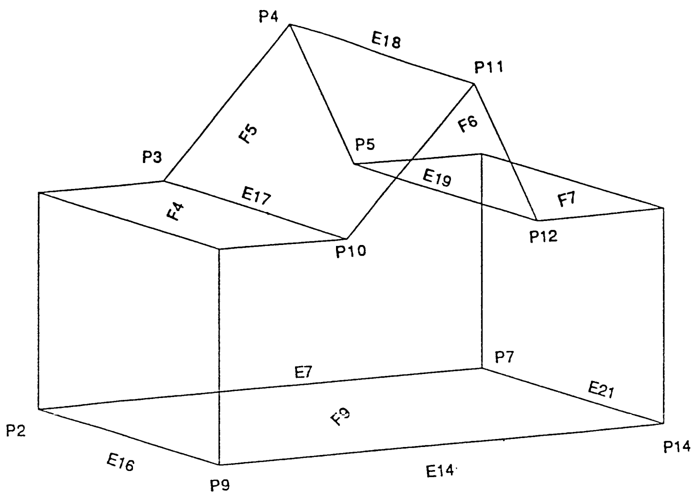


Figure 2a. CAD model of the object A.

0	ENDTAB	CONTINUOUS
SECTION	0	0
2	TABLE	LAYER
TABLES	2	2
0	LTYPE	AME_FRZ
TABLE	70	70
2	1	65
VPORT	0	62
70	LTYPE	7
2	2	6
0	CONTINUOUS	CONTINUOUS
VPORT	70	0
2	64	ENDTAB
*ACTIVE	3	0
70	Solid line	TABLE
0	72	2
10	65	STYLE
0.0	73	70
20	0	1
0.0	40	0
11	0.0	STYLE
1.0	0	2
21	ENDTAB	STANDARD
1.0	0	70
12	TABLE	0
5.5	2	40

Figure 2b. AUTOCAD output in DXF format for the object A.

3.1.1. *Extraction of Geometrical and Topological Information of the Object*

The 3D model of the objects are created using AUTOCAD with advanced modelling extensions (AME). In AUTOCAD(AME), bodies are created using standard shapes, such as cubes, blocks, wedges, cones, cylinders, etc. A complex body is constructed by binary operations such as union, substraction or intersection between the bodies already defined. Transformations can be applied to a body by defining translation, rotation or scale data. The geometric data of the object obtained at the output of the CAD system is in DXF format, consisting of all the faces, edges and vertices of the object. Figure 2(b) shows an example of the DXF output for the sample object shown in Figure 2(a). It can clearly be seen that these data need to be postprocessed to be useful in the object recognition process. The postprocessing of the data is done as explained below.

3.1.2. *Calculation of Stable Positions and Corresponding Views of the Object in its Stable Positions*

We have already mentioned that during the run-time mode the object is presented to the system lying on a worktable in one of its stable poses. Hence, after modelling an object, it is necessary to determine all its stable poses and a list of potentially visible features that will appear in the eye of the overhead camera in each of these stable poses. A 3D planar object model has a finite number of stable positions. This is a direct consequence of the fact that the object is converted into a polygonal form by interpolation of the curved edges their replacement by piecewise linear segments. The curved faces are transformed into pieces of planar faces by joining the corresponding points representing the curved edges. The stable poses of an object can be determined by first calculating the 3D convex hull of the object. The convex hull is a polyhedron containing among its faces all the object's convex faces. A convex face is such that the object must lie on one side of the plane defined by this face. The problem of finding the stable poses of the object then reduces to one of determining the stable positions of the convex hull. Each face of the convex hull is constrained to lie flat on the worktable and this constraint determines a potential stable position. The stability of a pose is verified by projecting the centre of gravity of the object down onto the worktable and checking whether it falls within the boundaries of the supporting face. The details of the algorithms employed for 3D convex hull and stable pose generation can be found in [18].

Once the stable positions generated are verified, for each stable position, the coordinates of all the object vertices are redefined using linear and rotational transformations so that the horizontal worktable plane ($Z = 0$) coincides with the face on which the object rests and the origin $(0, 0, 0)$ lies at one of the vertices of this face. A list of faces visible in the perspective view is then determined for each stable position. For a particular pose, if a face is parallel to the worktable plane it may be fully visible. If the face is inclined at an angle to the worktable

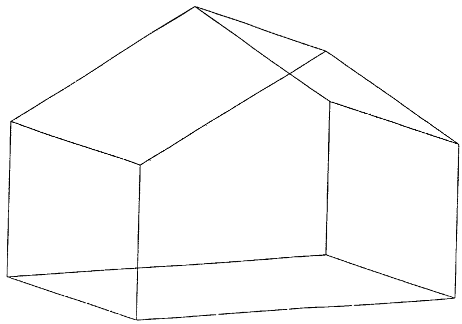


Figure 3a. Convex hull of the object A.

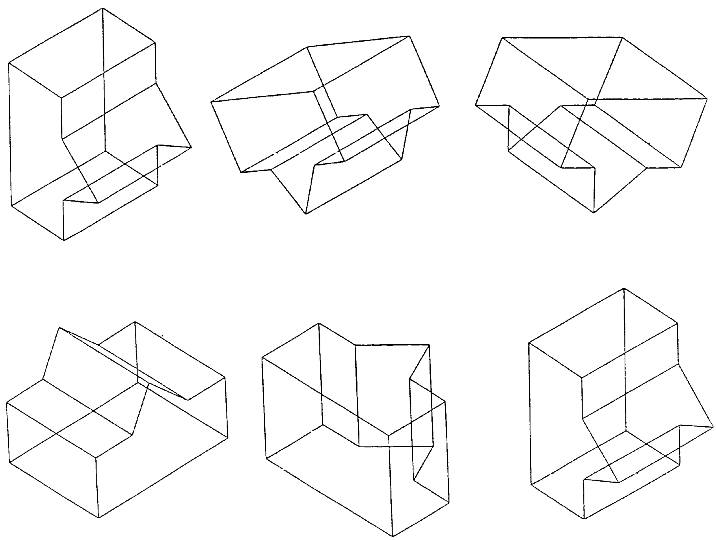


Figure 3b. Stable poses of the object A.

plane, only a part may be visible while if a face is perpendicular to the worktable plane it may not be visible at all. Once the list of fully and partly visible faces are available, the next step is to form the perspective view boundary, i.e., to locate all the visible edges, visible vertices, their order of connections and faces they correspond to and thereafter calculate a list of visible features. Figure 3(a) shows the convex hull and Figure 3(b) shows the different stable poses of the sample object shown in Figure 2(a).

3.1.3. *Determination of Visual Features*

For a given stable pose of an object, the total number of faces which are fully and partly visible and the angle between the faces are determined. The set of edges that bound the visible faces are determined and are classified into boundary and

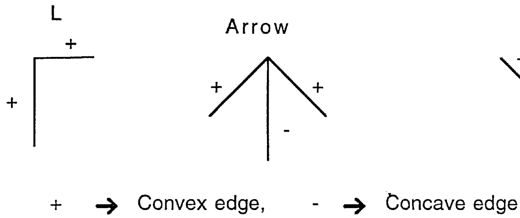
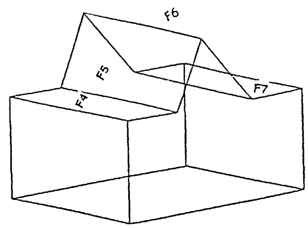


Figure 4. Junction terminologies.



NAME OF THE OBJECT FILE : OBJ5.FEA

VISUAL FEATURES EXTRACTED FROM CAD MODELER

Stable Pose : 2
Translation Vector : [-9.000000 -2.000000 -5.000000]
Rotation Matrix :

$$\begin{bmatrix} +0.000000 & +0.000000 & -1.000000 \\ -1.000000 & +0.000000 & +0.000000 \\ +0.000000 & +1.000000 & +0.000000 \end{bmatrix}$$

Centroid : [2.500000 3.477700 1.785100]
Total No. of Vertices on the Worktable : 4
Original Points : 4
: [P2 P7 P9 P14]
Interpolated Points : 0
Number of Edges on the Worktable : 4
Original Edges : 4
: [E7 E14 E16 E21]
Interpolated Edges : 0
Total No. of Faces on the Worktable : 1
: [F9]

TOTAL No. OF FULLY VISIBLE FACES : 4 (F4,F5,F6,F7)
PARTIALLY VISIBLE FACES : 0
VERTICAL FACES : 4
HIDDEN FACES : 0

FACE INFORMATION
Face Name : (F4,F5)
Type of Interaction : CONCAVE EDGE
Degree of Concavity / Convexity : 0.707107
Angle Between Faces : 45.0000
Face Name : (F5,F6)
Type of Interaction : CONVEX EDGE
Degree of Concavity / Convexity : 0.316228
Angle Between Faces : 71.5651
Face Name : (F6,F7)
Type of Interaction : CONCAVE EDGE
Degree of Concavity / Convexity : 0.447214
Angle Between Faces : 63.4349

EDGE INFORMATION
Total No. of Visible Edges : 13
Boundary Edges : 10
Non Boundary Edges : 3

ADJACENCY RELATIONSHIP
Edge Name : (E17)
Edge Type : CONCAVE
Start Vertex : Arrow Junction [P3]
End Vertex : Arrow Junction [P10]
Left Face : F4
Right Face : F5

ADJACENCY RELATIONSHIP
Edge Name : (E18)
Edge Type : CONVEX
Start Vertex : Fork Junction [P4]
End Vertex : Fork Junction [P11]
Left Face : F5
Right Face : F6

ADJACENCY RELATIONSHIP
Edge Name : (E19)
Edge Type : CONCAVE
Start Vertex : Arrow Junction [P5]
End Vertex : Arrow Junction [P12]
Left Face : F6
Right Face : F7

VERTEX INFORMATION
Total No. of Visible Vertices : 10

JUNCTION INFORMATION
Face Name : (F4)
No. of Edges : 4
Junction Type :(L(2),A(2))

Face Name : (F5)
No. of Edges : 4
Junction Type :(A(2),F(2))

Face Name : (F6)
No. of Edges : 4
Junction Type :(F(2),A(2))

Face Name : (F7)
No. of Edges : 4
Junction Type :(A(2),L(2))

Figure 5. Features extracted from CAD model of object A.

non-boundary edges. A boundary edge separates an object from the background. A non-boundary edge separates two neighbouring faces. The non-boundary edges are again classified as concave or convex. An edge separating two faces is called concave if it is towards the viewer, whereas it is called convex if it recedes away from the user's view point. The concave/convex edges are further classified based on their degree of concavity/convexity. The degree of concavity/convexity is determined by calculating the dot product of the surface normals of the respective faces. Points where two or three edges meet is called a junction. Three different types of junctions are considered in this work namely L, FORK and ARROW type junctions. Figure 4 shows the junction terminologies used. Figure 5 shows the features extracted for the sample object in the stable pose shown in Figure 2(a).

3.1.4. *Representation of the Relational Graph Corresponding to Each View of the Object in its Various Stable Poses*

The object in every stable pose is represented as a graph in which the nodes correspond to junctions and the arcs connecting two nodes correspond to an edge separating two faces. Nodes carry information as to whether they represent an L, FORK or ARROW junction. Arcs carry information as to whether the edge they represent is a boundary/non-boundary edge and, if it is a non-boundary edge, whether a concave/convex edge, along with the degree of concavity/convexity. The shortest circuit in which a boundary edge participates determines a visible face of the object. This representation gives the relational graph of the view of the object in its current stable pose. Figure 6 shows such a representation for the sample object of Figure 2(a).

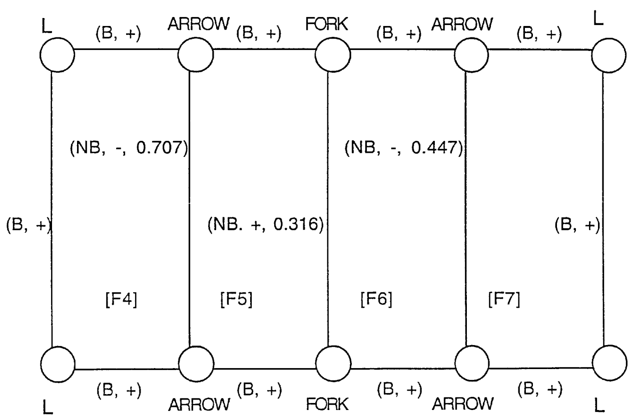


Figure 6. Relational graph created for the object A.

3.2. EXTRACTION OF IMAGE FEATURES

In the run-time mode a similar relational graph structure is extracted from two perspective views of the object, taken from two different viewpoints with parallel viewing directions. The two images are analysed to extract lines and a connected topographic structure of the view. For each visible face, the corresponding line sets and their vanishing points are identified. The 3D direction of the lines are related to their vanishing points by a simple relation. The cross-product of the 3D directions of any two non-parallel lines lying on a face determines the face normal. This information is then made use of in determining various other invariant features and is incorporated into the relational graph structure representing the view of the object. More details of this algorithm are available in [20].

The above procedure consists of the steps described in Sections 3.2.1–3.2.6.

3.2.1. *Acquiring Perspective Images*

Two perspective images of the object are taken by moving the camera a small distance along the viewing direction.

3.2.2. *Edge Detection*

Canny's edge detector [21] is used to obtain all the edges in the two images.

3.2.3. *Grouping Edge Pixels into Lines*

The detected edge points are grouped into various straight lines using the Hough transform approach. Figure 7 shows the results of image processing for the object shown in Figure 2(a).

3.2.4. *Junction Formation*

Since trihedral objects are considered, junctions in the 2D views are formed by two or three lines. The junction coordinates are determined as follows: for two line junctions, the coordinates of the junction point is obtained as the intersection point of the two lines. For three line junctions, the junction coordinates are the average of the coordinates of the three end points of the lines forming the junction.

3.2.5. *Generation of Circuits*

The lines are grouped into circuits where each circuit corresponds to a different visible face. This is done by employing a brute force algorithm for generating all the circuits formed by the line data. A non-boundary edge can participate in only one circuit. Hence the generated circuits are pruned by retaining only the shortest circuits in which non-boundary edges participate.

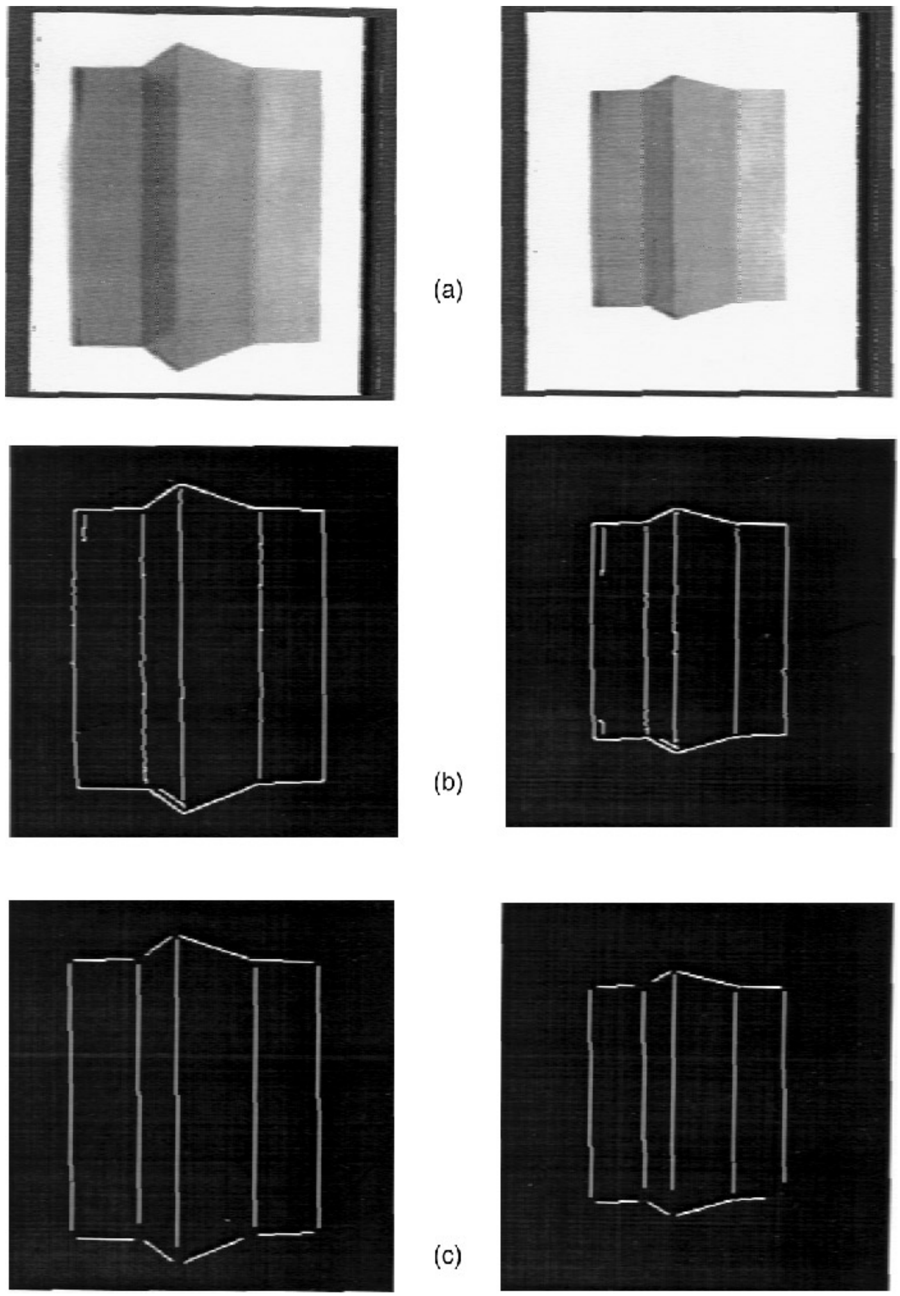
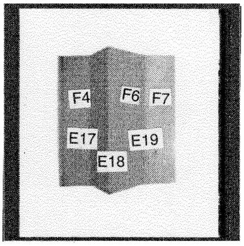


Figure 7. Image processing results for object A.



Feature from perspective images

Face information

- Total number of visible faces : 4

Face interaction : {F4, F5}

- Type of face interaction : Concave
- Degree of concavity : 0.652
- Angle between faces : 51 Deg

Face interaction : {F5, F6}

- Type of face interaction : Convex
- Degree of concavity : 0.440
- Angle between faces : 65 Deg

Face interaction : {F6, F7}

- Type of face interaction : Concave
- Degree of concavity : 0.441
- Angle between faces : 64 Deg

Edge information

- Total number of visible edges : 13
- Number of boundary edges : 10
- Number of non boundary edges : 3

Vertex information

- Total number of visible vertices : 10

Junction information

- L Junction (4)
- Fork junction (2)
- Arrow junction (4)

Figure 8. Output of computer vision process for object A.

3.2.6. *Extraction of Features*

The primary feature which is extracted from the images is the face normal of each visible face. For this, the different lines in view 1 and view 2 are caused to correspond. This line correspondence is done in a hierarchical fashion by first corresponding the faces based on ordering of X, Y coordinates of the centre of each face and then by corresponding the lines in each face by ordering the X, Y coordinates of their end points. Corresponding lines in the two views can be considered as the projection of parallel lines in space. These lines meet at points called vanishing points [19]. Hence, for each visible face, corresponding line sets and their vanishing points are computed. The vanishing points for two sets of parallel lines which are not parallel to each other, lying on the same face are then considered for computing the face normal. The 3D direction of the lines is related to their vanishing points by a simple relation [20]. If the 3D directions of the two non-parallel lines lying on a face are computed as (a, b, c) and (d, e, f), then the face normal is given by the cross-product of (a, b, c) and (d, e, f). Once the normal to the visible faces are known, invariant features, e.g., concave/convex edges, degree of concavity/convexity, angle between the faces, junction type etc. are computed either from view 1 or view 2. This information is then used to determine the relational graph structure of the view of the object. Figure 8 shows the results of vision processing for the sample object in one of its stable poses as shown in Figure 2(a).

3.3. MATCHING MODEL DATA WITH IMAGE DATA

Matching uses a hierarchical decision tree, where different subset of features are used at various decision levels. The steps for the design of decision tree are as follows:

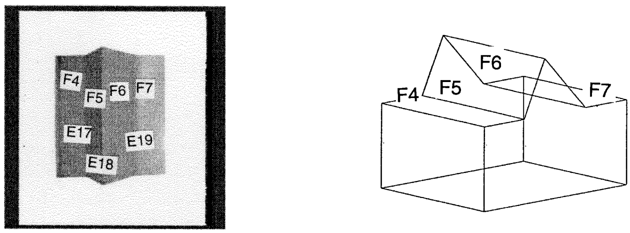
1. Extract a set of effective features for each model at each of its pose.
2. Classify the features into various logical levels based on their computational complexity and dependency on other features. Independent features are assigned logical level 0, whereas each succeeding level contains increasingly more complex features.

   In the present problem, features such as a list of edges may belong to logical level 0, since they are obtained from the raw image data after preprocessing. Features such as classification of edges as boundary and non-boundary, list of visible faces, start and end vertex of each edge may belong to logical level 1 since they can be calculated only after obtaining the edge list for logical level 0.

   Similarly other more complex features, such as an edge separating two faces is concave or convex and angle between the two faces belong to logical level 2 etc.
3. Construct the decision tree by first selecting the best feature (most robust, low cost belonging to logical level 0) as the root node of the tree. Two children of the root node are determined and all models that have feature value less than a specified threshold are associated with the left child, whereas the right branch is assigned all models with their feature value greater than or equal to the threshold. This process is repeated and the newly generated left and right node at every level is assigned features from logical level 0 until all the features of that level are exhausted or there is no feature left in that level which can distinguish the given set of models.

The tree generation now continues by selecting features from the next logical level. The process continues at all successive logical levels till the leaf nodes of the tree are reached. Each of these leaves uniquely determine a given model at a given pose. The tree is constructed during the off-line mode of the system.

In the run time mode, matching started from the root node and only those features as directed by the decision tree are extracted. Therefore, in order to identify a model at a given pose, it is not necessary to extract all the features. This reduces the computation time to a considerable extent and the matching is done in a reasonable amount of time.



(a) (b)

Feature from perspective images		Feature from CAD modeller	
Face information		*Face information*	
- Total number of visible faces	: 4	- Total number of visible faces	: 4
Face interaction : {F4, F5}		Face interaction : {F4, F5}	
- Type of face interaction	: Concave	- Type of face interaction	: Concave
- Degree of concavity	: 0.652	- Degree of concavity	: 0.707
- Angle between faces	: 51 Deg	- Angle between faces	: 45 Deg
Face interaction : {F5, F6}		Face interaction : {F5, F6}	
- Type of face interaction	: Convex	- Type of face interaction	: Convex
- Degree of concavity	: 0.440	- Degree of concavity	: 0.316
- Angle between faces	: 65 Deg	- Angle between faces	: 71.6 Deg
Face interaction : {F6, F7}		Face interaction : {F6, F7}	
- Type of face interaction	: Concave	- Type of face interaction	: Concave
- Degree of concavity	: 0.441	- Degree of concavity	: 0.447
- Angle between faces	: 64 Deg	- Angle between faces	: 63.4 Deg
Edge information		*Edge information*	
- Total number of visible edges	: 13	- Total number of visible edges	: 13
- Number of boundary edges	: 10	- Number of boundary edges	: 10
- Number of non boundary edges	: 3	- Number of non boundary edges	: 3
Vertex information		*Vertex information*	
- Total number of visible vertices	: 10	- Total number of visible vertices	: 10
Junction information		*Junction information*	
- L Junction (4)		- L Junction (4)	
- Fork junction (2)		- Fork junction (2)	
- Arrow junction (4)		- Arrow junction (4)	

Figure 9. Comparison of CAD and vision data for object A.

4. Results

The algorithm proposed in this paper has been implemented and tested for a set of sample planar objects. The objects are modelled using the AUTOCAD(AME) modeller available in PC/AT. For the processing of vision data, a SUN 4/280 system with a frame grabber card is used. Both the CAD and the vision data are processed using the above-mentioned procedures. Figure 9 shows a comparision of the results obtained using the CAD and the vision data for the sample object shown in Figure 2(a).

5. Conclusions

We have proposed a system for 3D planar object recognition using their CAD models. The generation of computer representations for the object models are obtained in a systematic manner, from their already available CAD models, without human intervention. This makes the system more flexible, fast and efficient than conventional modelling approaches wherein modelling is done by hand or by training from images. The run-time vision process is a novel approach where the face normals which is an important feature, has been used for object recognition. The overall scheme has given reasonably accurate results and can be used in fully automated industrial environments, for a variety of applications.

Acknowledgements

The authors wish to thank Sri. G. Kannan, M. Tech. student, Anna University, Madras, for the validation of the code for object database generation from the CAD models. The authors express their sincere gratitude to Sri. L. Umesh for his help in the preparation of the manuscript. The authors also thank Director CAIR, Bangalore for his kind permission to publish this paper.

References

1. Chin, R. T. and Dyer, C. R.: Model-based recognition in robot vision, *Computing Surveys* **18**(1) (1986).
2. Filkins, B. L.: Graphical concepts in image processing – a bridge between two worlds, *SPIE on Applications of Digital Image Processing* **575** (1985), 18–23.
3. Roberts, L. G.: Machine perception of three dimensional solids, in: J. T. Tipett et al. (eds), *Optical and Electro-optical Information Processing*, MIT Press, Cambridge, MA, 1965.
4. Brooks, R. A.: Simbolic reasoning among 3D models and 2D images, *Artificial Intelligence* **17** (1981), 285.
5. Lowe, D.: *Perceptual Organisation and Visual Recognition*, Kluwer Academic Publishers, Boston, MA, 1985.
6. Baumgart, B. G.: Geometric modelling for computer vision, Technical Report AIM-249, STANCS-74-463, Computer Science Dept., Stanford University, 1974.
7. Hermen, M.: Representation and incremental construction of a three-dimensional scene model, CMU Report, CMU-CS85-103, 1985.
8. Koshikawa, K. and Shirai, Y.: A 3D modeller for vision research, in: *Proceedings of the International Conference on Advanced Robotics*, Tokyo, Japan, 1985.

9. Bolles, R. C. and Horaud, P.: 3DPO: A three dimensional part orientation system, *Robotics Research* **5**(3) (1986).
10. Henderson, T., Hansen, C., Samal, A., Ho, C. C., and Bhanu, B.: CAGD-based 3D visual recognition, in: *Proceedings ICPR*, Paris, 1986, pp. 230–232.
11. Bhanu, B. and Ho, C. C.: CAGD-based 3D object representations for computer vision, *IEEE Computers* **20**(8) (1987), 19–36.
12. Ho, C. C.: CAGD-based 3D object representations for computer vision, Master's Thesis, University of Utah, Salt Lake City, UT, 1987.
13. Flynn, P. J. and Jain, A. K.: CAD based computer vision – From CAD models to relational graphs, *IEEE PAMI* **13**(2) (1992).
14. Narasimhamurthi, N. and Jain, R.: CAD based object recognition incorporating metric and topological information, *SPIE* **938** (1988), 436–443.
15. Merat, F. L. and Radack, G. M.: Automatic inspection planning within a feature based CAD system, *Robotics and Computer Integrated Manufacturing* **9**(1) (1992), 61–69.
16. Park, H. D. and Mitchell, O. R.: CAD based planning and execution of inspection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, 1988, pp. 858–863.
17. Majumdar, J., Levi, P., and Rembold, U.: 3D model based robot vision by matching scene description with the object model from a CAD modeller, *Robotics and Autonomous Systems* **5** (1989).
18. Majumdar, J., Levi, P., and Rembold, U.: Algorithm for visible surface pattern generation – a tool for 3D object recognition, in: *Sensor Based Robots*, NATO ASI Series, Vol. F-66, 1991, pp. 93–106.
19. Magee, M. J. and Aggarwal, J. K.: Determining vanishing points from perspective images, *Computer Vision, Graphics and Image Processing* **26** (1984), 256–267.
20. Majumdar, J., Rangarajan, K., and Seethalakshmy, A. G.: Computation and use of planar face normals, *Pattern Recognition Letters* **14**(10) (1993), 809–816.
21. Canny, J. F.: A computational approach to edge detection, in: M. A. Fishler and O. Firschein (eds), *Readings in Computer Vision*, Morgan Kaufmann, 1987.