Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C., Canada • October 1998

Robotic Vision: 3D Object Recognition and Pose Determination

A. K. C. Wong, L. Rong and X. Liang
Pattern Analysis and Machine Intelligence Group
University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

1 Introduction

1.1 Motivations and Objectives

A challenge in 3D computer vision is to automatically acquire 3D models of objects through a CCD camera and to use the acquired models to recognize objects and estimate their poses. From the practical standpoint, such approach is more flexible and cost effective than the use of laser range scanners [10, 11]. Nevertheless, some major hurdles have to be overcome in order to reduce the overhead of this approach. The objective of this paper is to overcome them.

Object recognition involves the identification and pose estimation of one or more objects in a scene using a previously constructed database of models. Excellent reviews can be found in [8, 1]. Today, most systems attempt to deal with incomplete data, occlusions, and various forms of image degradations. However, some important and critical issues related to the feature extraction and model matching such as robustness, speed, efficiency and practicality have not been well addressed. This paper describes an efficient and fast feature extraction and pose determination system for real time object recognition.

1.2 PAMI Monocular 3D Vision System

The PAMI System is based on our early work [10, 11] and those in [2]. A description of the entire system, including both the model synthesis component and the object recognition component, is shown in Figure 1. This paper focuses on the object recognition component.

The PAMI System works on images acquired from a single CCD camera [10]. It first detects salient features from an image and then groups them according to their types as well as their spatial, geometrical and topological relations. The feature grouping types include: a) four corner points and triplets of lines forming corners; b) curve segments fitted into ellipses. The use of matching hypotheses generated based on feature groupings is usually more robust and effective than the combinatorial matching of point features. Its theoretical basis is derived from Fischler and Bolles' algorithm [4] and PAMI's 3-Point Pose Algorithm. It determines the 3D object pose from a 2D image by relating the feature groupings with their corresponding model features. Once the correspondence is established, object model with the detected object pose is back-projected onto the 2D image plane for confirmation. The closeness of the back-projection frame to the image features is evaluated by two measures: "plausibility" and "reliability". The first helps to determine if the object is successfully recognized. The second is for fine-tuning the pose. To resolve instability and sensitivity to noise, we exploit data redundancies for verification and refinement of the initial recognition result.

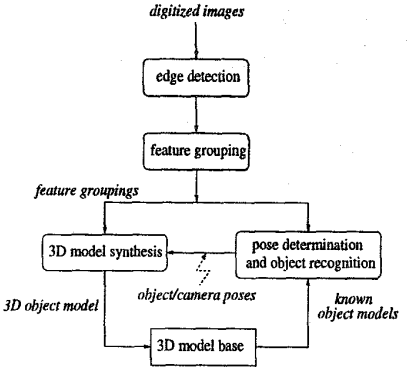


Figure 1: System configuration of PAMI Monocular 3D Vision System

The detailed functional modules of our 3D object recognition system is described in Figure 2. The processing level of information of the system is subdivided into three main blocks: a) 2D feature detection, b) 2D feature grouping and c) pose determination and object recognition. We first describe our feature extraction and grouping algorithms, then the pose determination procedure and finally the experiments to render system performance evaluations and robotic applications.

0-7803-4465-0/98 $10.00 © 1998 IEEE

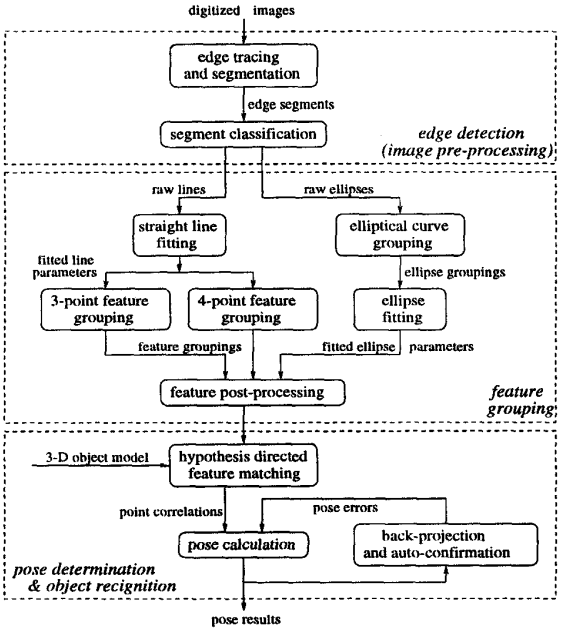


Figure 2: The software modules of monocular 3D object recognition and pose determination system

2 Feature Extraction

Our feature extraction procedure consists of: a) edge detection and b) feature grouping.

2.1 Edge Detection

The edge detection method, adopted from [5], uses a set of curve partitioning and grouping rules based on the perceptual organization of descriptive curve features. It tracks curve segments and joins them into an appropriate form of a curve structure according to its topological and geometrical properties. To overcome an disadvantage in [5] which merges different traces not belonging to the same curve together, geometrical criteria are introduced to segment the merged traces.

2.1.1 Edge Trace Segmentation

The edge detector provides edge traces on which each point is described by four attributes: a) (X,Y): the coordinates of a point in the original image; b) *Transient*: a boolean indicating discontinuity of the trace at a point; c) *Slope*: slope obtained from the three-edge-point average; and d) *Curvature*: the change in slope of two adjacent trace segments.

If a point is on a smoothly connected trace, its transient should stay zero (i.e. unchanged). We consider the status of a point unstable if its transient is not zero. Thus, all corners, junctions, curve breaking points and local spurious noises are points with non-zero transient values. Edge traces separated at these points and the parts between them require further processing. Segments with lengths less than a threshold are considered to be noise and are removed.

2.1.2 Segment Classification

Once a segment is extracted from the edge trace, we classify it into line or elliptical curve according to the changes of its slope. To find the connection point, the curvature is used to assign segments into a line or a curve class. Points in adjacent segments of the same class are grouped and fitted into lines and ellipses.

2.2 Feature Grouping

Feature grouping clusters 2D features into groups each of which could be associated to an object to be recognized. This effectively reduces the search space for object recognition.

The Procedure: *Given a set of 2D features F, the grouping procedure C outputs a set of possible connected groups $\{F_i\}$ each of which can be related to one or more possible objects $\{obj_j\}$.* We include the constructed model of each object obj_j in the model base and restrict the elements of F to fitted straight lines or elliptical/circular curves.

2.2.1 Line Fitting and Grouping

To accurately locate a corner, we first fit the line segments forming that corner and then obtain the equation of each line. Here, we use $x/a + y/b = 1$ for line fitting where a is the *x-intercept* and b is the *y-intercept*. The *line slope* is computed as $-b/a$. We refer a horizontal and a vertical line as of Type 1 and of Type 2 respectively. Any other is of Type 3. From segment type, we can estimate its parameter(s) accordingly. If its $X-intercept$ a and $Y-intercept$ b are not equal to zero, we use the *median of intercepts* method to fit the line. If N points are in a line fitting set, then any two points i and j can form a line and give an (a_{ij}, b_{ij}) pair. There are at most $N(N-1)/2$ pairs each of which can be described as

$$a_{ij} = \frac{x_j y_i - x_i y_j}{y_i - y_j}, (y_i \neq y_j), \quad b_{ij} = \frac{x_i y_j - x_j y_i}{x_i - x_j}, (x_i \neq x_j) \tag{1}$$

where $1 \leq i \leq j \leq N$. a and b can be obtained by:

$$a = median\{a_{ij}\}, \qquad b = median\{b_{ij}\}$$

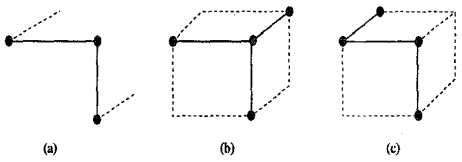


Figure 3: Types of feature groupings: (a) a 3-point L-shape feature; (b) a Y-shape 4-point feature grouping; (c) a concatenated 4-point feature grouping

We use the following rules to obtain corners from intersecting lines: a) a corner must be located at the extensions of all the lines in the set (to avoid having $T-junctions$); b) two parallel lines in a set do not form a corner; and c) if several intersections are found, their average position is taken as the corner position. Once the corners are obtained, it is easy to find groups of 3-point and 4-point features as shown in Figure 3.

2.2.2 Curve Grouping and Ellipse Fitting

The equation applied for ellipse fitting is:

$$\frac{[(x-a)\cos\theta+(y-b)\sin\theta]^2}{c^2}+\frac{[-(x-a)\sin\theta+(y-b)\cos\theta]^2}{d^2}=1 \quad (2)$$

where (a,b) is the center of the ellipse, c, d ($c>0$ and $d>0$) are its axes, and θ is its rotation angle (orientation) — the angle between the x axis and the short axis of the ellipse. When $c=d$, the ellipse becomes a circle.

Accurate estimation of the five basic parameters of an elliptical curve (the center coordinates, the major and minor axes, and the orientation) arises in various machine-vision related problems, mostly based on the local fitting concept. Due to the presence of short curve segments and local spurious noise, local fitting often yields hyperbolas or other non-elliptical curve forms. To avoid this, we introduce the global fitting [13], ie. before fitting a selected curve segment, the entire curve set is searched to see if it can be grouped to form an ellipse in compliance to their 2-D geometrical and topological constraints. Thus, the influence of noise and the possibility of fitting points from different curves are greatly reduced.

With appropriate curve attributes, we select a curve segment from the curve set and start searching the starting and ending points of its neighboring segments. Segments with one of their terminating points within the search range of an end point are taken as candidates to form corners if certain conditions of curve adjacency are satisfied.

To fit data to an ellipse is to find how close an ellipse equation could describe a set of curve data believed to be part of an ellipse. We use a fast iterative *Newton* method [13]. It starts with an initial solution for the parameters, fits the equation to the data iteratively until the error requirement is satisfied.

3 Object Recognition and Pose Determination

3.1 Hypothesis Directed Feature Matching

Recognizing an object from a 2D image could be considered as a feature matching process. It attempts to establish correspondence between feature groupings from 2D images and the 3D features from models in the object model base. In general, this is an NP-complete problem. To speed up matching, we introduce a hypothesis directed search based on [10]. Hypothesis generated from observed feature groupings are used as heuristics and are evaluated with the criteria: a) *plausibility*: a threshold to determine if the established matching is acceptable; and b) *reliability*: a measure based on the matching error and a weighting score from the feature grouping itself.

The algorithm:

1. generate a search tree from the available image feature groupings and 3D model features;
2. select an unmatched image feature grouping and a 3D model feature to establish a matching hypothesis;
3. estimate the characteristic view [3, 12] (a 2D description of the 3D object based on on a perspective projection) according to the hypothesis;
4. evaluate the plausibility and reliability of the matching hypothesis;
5. verify the hypothesis and record the successful matching;
6. go to step 2 to search for new matching pairs until all feature groupings has been exploited or a plausible/reliable matching has been found;
7. output the result.

3.2 Pose Determination

Pose determination, in both analytical closed form and numerical solutions, is a well studied problem

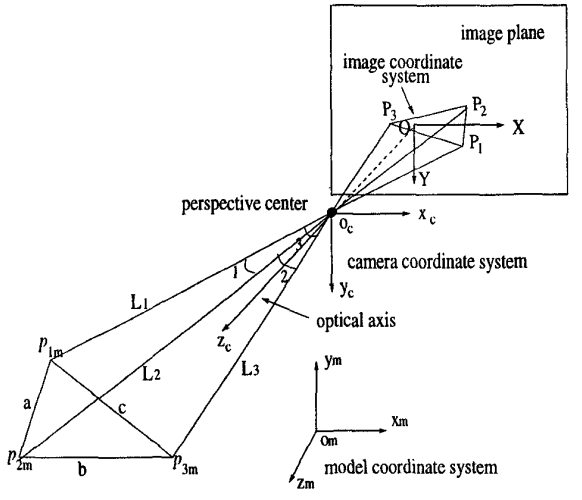


Figure 4: The three coordinate systems in the pose determination problem

[4, 6]. Among these are 3-point, 4-point and 6-point approaches. In applications, approaches with multiple solutions are preferable to those that generate an unique solution since redundant information from the image can be used to prune the solution space and multiple poses to improve the accuracy. Based on the 3-point multiple solution algorithm, we develop a pose determination algorithm to resolve the problems of speed, accuracy and uncertainty.

Let $(p_1, p_2, ..., p_n)$ be point features of the observed object in a coordinate system pre-assigned to the object and $(P_1, P_2, ..., P_n)$ be the corresponding perspective projection points on the image plane. A pose determination algorithm would yield a rotation matrix R and a translation vector T which together map $(p_1, p_2, ..., p_n)$ onto $(P_1, P_2, ..., P_n)$.

3.2.1 Geometry, Projections and Transformations

In pose estimation, three coordinate systems: image, camera and model system are adopted (Figure 4). We use the lower and upper cases, p's and P's respectively to refer to the points in the 3D space and on the 2D plane. The subscripts c and m signify that the point is expressed in the camera or model coordinate systems respectively.

Image Coordinate System:
The image coordinate system **I** is defined on the 2D image plane with the origin at the geometric center of the image and with an x axis and a y axis as horizontal and vertical axes respectively. To keep the image in a positive orientation we set the image plane perpendicular to the viewing axis at a distance f from the camera center.

Camera Coordinate System:
The camera coordinate system **C** is a viewer centered reference system such that the observer is located at the origin which is the center of the lens. The viewing axis is collinear to the z axis and the x and y axes are chosen to be collinear to the x and y axes of **I** respectively. In Figure 4, the point on the image corresponding to an arbitrary point in space is located at the intersection of the image plane and the line joining the point to the center of the lens. Thus, if a point has coordinates $(x, y, z)_c$ in the camera system, its image point P will be $(X, Y, f)_c = (\frac{xf}{z}, \frac{yf}{z}, f)_c$.

Model Coordinate System:
The model coordinate system **M** is used to describe the CAD model of known objects, usually with the origin selected at the center or at one of the object's corners while the z axis is perpendicular to a surface plane.

3.2.2 The Perspective Tetrahedron

Suppose that we have image points P_1, P_2, P_3 and their 3D counterparts p_1, p_2, p_3. According to the perspective projection, the positions of p_1, p_2 and p_3 are determined from the image rays through the corresponding image points P_1, P_2 and P_3.

A solution set of L_1, L_2 and L_3 in Figure 4 can be obtained by applying cosine law as below:

$$R_{12}^2 = L_1^2 + L_2^2 - 2L_1L_2cos\theta_{12} \quad (3)$$

$$R_{13}^2 = L_1^2 + L_3^2 - 2L_1L_3cos\theta_{13} \quad (4)$$

$$R_{23}^2 = L_2^2 + L_3^2 - 2L_2L_3cos\theta_{23} \quad (5)$$

where R_{12}, R_{13} and R_{23} are the three inter-point distances among the three object points: $R_{12} = |p_2 - p_1|$, $R_{13} = |p_3 - p_1|$ and $R_{23} = |p_2 - p_3|$.

We apply the algorithm [4] to obtain the three angles and transform the above equations to a quartic polynomial whose roots are solved numerically.

3.2.3 Rotation Matrix and Translation Vector

Let R be the rotation matrix; T be the translation vector from **C** to **M**; p_{1m}, p_{2m} and p_{3m} be the coordinates of the three object points in **M** and p_{1c}, p_{2c} and p_{3c} be the coordinates of the three object points in **C**. p_{1c}, p_{2c} and p_{3c} can be determined from their locations in **I** by:

$$p_{im} = Rp_{ic} + T, \qquad i = 1, 2, 3. \quad (6)$$

With the known locations of p_{1m}, p_{2m} and p_{3m} in **M** and L_1, L_2 and L_3 from the solution, T can be solved by a set of three quadratic equations. Then we can transform Equation (6) into:

$$p_{im} - T = Rp_{ic} \quad i = 1, 2, 3 \tag{7}$$

Now we combine the three point vectors into a matrix and obtain:

$$\bar{p}_m = (p_{1m}, p_{2m}, p_{3m}) \text{ and } \bar{p}_c = (p_{1c}, p_{2c}, p_{3c}) \tag{8}$$

Then Equation (7) becomes:

$$\bar{p}_m - T = R\bar{p}_c \tag{9}$$

and we can solve R by:

$$R = \bar{p}_m \bar{p}_c^{-1} \tag{10}$$

3.3 Pose Verification and Refinement

The above pose estimation algorithm yields several solutions. The reliability and plausibility measures are then used as criteria for pose verification and refinement. Given one of the pose solutions R and T (called a "candidate"), it is straightforward to obtain the inverse transformation R' and T' which are then used to project the entire object model from **M** to the 2D image. In **I**, the back-projection error is then calculated based on the discrepancy between the 2D image features and the corresponding projected 3D model features. The back-projection error, the number of matched feature groupings and their pre-calculated weighting scores are then combined into a plausibility criterion. The candidate with the largest plausibility is selected as the best solution. If the measure of the best solution is still smaller than a pre-set threshold, we conclude that the object recognition fails. Factors such as image noises and the optical distortion of the camera lens may cause inaccuracies in pose estimation. The calculation of rounding error plays an important role for the pose accuracy.

To enhance the reliability, we introduce a pose refinement process to "fine-tune" the solutions using the redundant information. In the algorithm, R and T are calculated by three pairs of matching points. When extra matching feature pairs and ellipse/circle pairs are available, they will be used only for evaluating the matching errors and the plausibility criterion. In pose refinement, among all the matching pairs, a combination of four pairs with the least uncertainty and largest plausibility are selected for pose re-calculation. The new "fine-tuned" pose result would in general agree with the initial one but with higher accuracy and with ill-condition problem avoided.

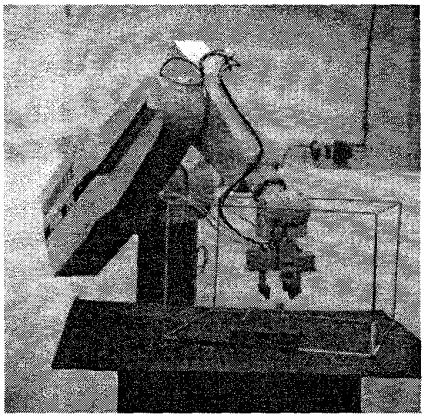


Figure 5: The PAMI intelligent robot system: a CCD camera mounted on PUMA760 and objects and frame posed in arbitrary pose; guided by the vision system, the manipulator moves in to engage the object while avoiding the frame.

4 Implementation and Experimental Results

4.1 System Configuration

In the first set of experiments for system performance evaluation, the CCD camera is mounted on a tripod, sighting objects placed on a mobile platform. In the second set of experiments on vision based autonomous robotics, the camera is mounted near the end effector of a robot arm (Figure 5) to direct online automatic object manipulation and collision avoidance for an intelligent autonomous robotics project, supported by the Canadian Space Agency.

Our system is implemented on a Sun Ultra-Sparc, a Sun Sparc 330 and a PUMA 760 robot arm mounted with a CCD camera. The robot's task is to: 1) recognize and determine the poses of objects and the corner landmark of an aluminum frame using a CCD camera mounted on its end effector; 2) move its end-effector into the aluminum frame while avoiding collision with the frame; 3) engage the objects, one at a time, and transport them out of the frame. The vision system provides the robot with spatial information of the recognized object and the frame. Through an *eye-hand coordination* algorithm, the intelligent robot system relates the spatial information of the objects and the frame to the end effector. An on-line path planning algorithm then plans a trajectory through inverse kinematics and controls the movements of the robot arm

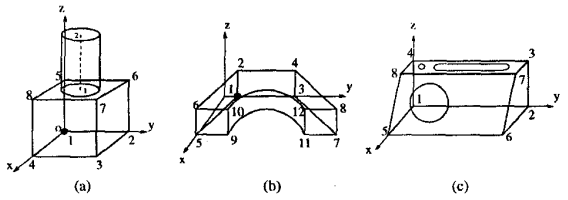


Figure 6: The CAD models for the 3D objects: (a) grenade, (b) bridge and (c) pen-holder respectively

Table 1: Percentage of Correct Recognition with Controlled Background

Objects	*1st sighting*	*2nd sighting*	*sighting>2*	*fail*
grenade	78.33%	11.67%	3.33%	6.67%
bridge	65.00%	21.67%	10.00%	3.33%
pen holder	60.00%	6.67%	20.00%	13.33%
average	67.78%	13.33%	11.11%	7.78%

to accomplish the task [9].

4.2 Experiments for Performance Evaluation

Two series of experiments are run for performance evaluation of the vision system. In each, three objects, an aluminum *grenade*, an aluminum *bridge* and a wooden *pen-holder* are used. Their CAD models are shown in Figure 6. Their conspicuous features include corners and complete and/or partial circles.

In the first series, the objects are positioned on a platform with controlled background but ambient lighting in a normal indoor environment. For each object, we conduct a total of 30 trials of object recognition in 3 groups (10 in each). The object is randomly rotated about its x, y and z axis respectively in the first, second and the third groups (Figure 6).

Table 1 lists experimental results. Each entry record the percentage of successful recognition out of 30 trials when success or failure was accounted. The sightings other than the first would automatically triggered when back-projection test detects failure in the previous run. If the object cannot be recognized even after five sightings, we consider that the recognition fails (fifth column). Figure 7 shows some samples on the back projection result. Figure 7 (a) through (c) illustrate successful recognitions of the grenade, the bridge and the pen-holder respectively.

In the second experiment series, the objects are positioned in a cluttered environment with other unrelated objects and uncontrolled illumination. We place each object in random positions when the images are taken. A total of 60 scenes are used for the experimentation. For each sighting, recognition on each of the three objects are tested. The rates of recognition are shown in Table 2 like those in the first experiment series. Due to the presence of clutters, the successful recognition rates with first sighting are dropped while the rates with multi-sightings are increased. Figure 8 illustrates some recognition results.

Table 2: Percentage of Correct Recognition in Cluttered Environment

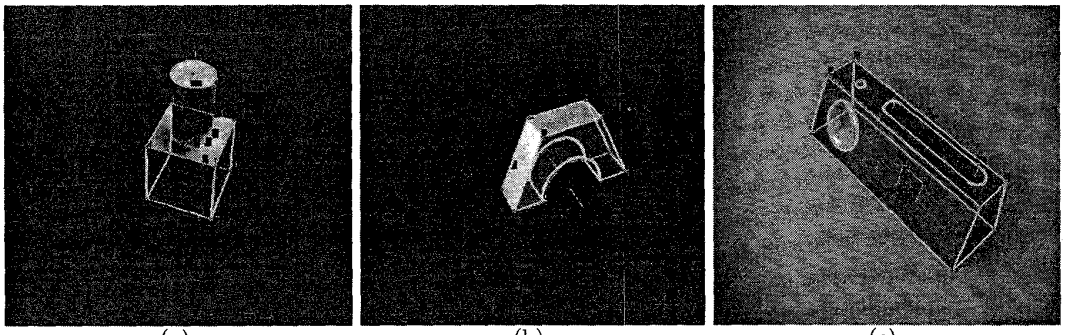
Objects	*1st sighting*	*2nd sighting*	*sighting>2*	*fail*
grenade	56.67%	23.33%	10.00%	10.00%
bridge	50.00%	16.67%	23.33%	10.00%
pen holder	43.33%	20.00%	23.33%	13.33%
average	50.00%	20.00%	18.89%	11.11%

4.3 Vision Based Autonomous Robotics Experiments

This experiment demonstrates a vision guidance vision robot system. Figure 9 shows the final result as all objects involved were confirmed by back projection of the CAD models. The objects include the *grenade*, the *bridge*, the *robot cap* and the *landmark* of the aluminum frame (Figure 5). This estimated spatial information is used: 1) to determine the free space and 2) generate a trajectory via inverse kinematics to move the end-effector into the frame as it takes the grenade and the bridge out of the frame, one at a time. Most of the time, the system's is robust under ambient lighting condition.

5 Conclusions

In conclusion, we observe that the prominence of our methodology lies in a) the use of spatial and topological feature groupings and b) an automatic pose verification and refinement algorithm. The vision system is integrated into an intelligent robot system capable of performing general vision guided tasks. Our future work is to further enhance the robustness of the vision system in recognition rate and pose estimation accuracy. Furthermore, this system has also been integrated with an autonomous guided vehicle (AGV) and an intelligent 3D vision and modeling system is under development to enable self-guiding navigation in constructing simple world environments.



(a) (b) (c)

Figure 7: 3D recognition results on objects: (a) grenade, (b) bridge, (c) pen-holder

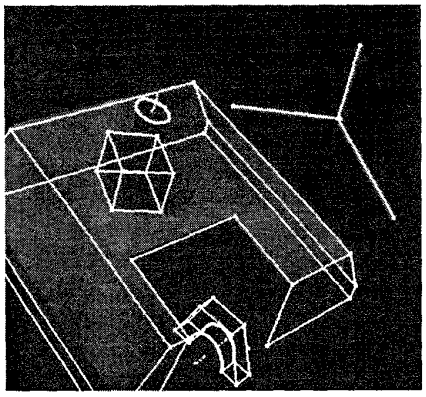


Figure 9: The back-projection of the recognized objects and the landmark target

References

[1] F. Arman and J. K. Aggarwal, "Model-Based Recognition in Dense-Range Images - A Review", *ACM Computing Survey*, Vol. 25, No. 1, pp. 67-108, 1993.

[2] P. J. Besl and R. C. Jain, "Three-dimensional Object Recognition", *Computing Survey*, Vol. 17, No. 1, pp. 74-145, 1985.

[3] I. Chakravarty and H. Freeman, "Characteristic Views as a Basis of Three-Dimensional Object Recognition", *Proc. of SPIE 336 (Robot Vision)*, pp. 37-45, 1982.

[4] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting Applications to Image Analysis and Automated Cartography", *ACM Communications*, Vol. 24, pp. 381-395, 1981.

[5] Q. C. Gao and A.K.C. Wong, "A Curve Detection Approach Based on Perceptual Organization", *Pattern Recognition*, Vol. 26, No. 7, pp. 1039-1046, 1993.

[6] R. M. Haralick, "Pose Estimation From Corresponding Point Data", *IEEE Trans. on SMC*, Vol. 19, No. 6, pp. 1426-1446, 1989.

[7] T. S. Huang, A. N. Netravali and H. H. Chen, "Motion and Pose Estimation Using Algebraic Methods", *Time-Varying Image Processing and Moving Object Recognition*, Cappellini, Ed., Amsterdam, The Netherlands: Elsevier, pp. 243-249, 1990.

[8] D. Nitzan, "Three-Dimensional Vision Structure for Robot Applications", *IEEE Trans. on PAMI*, Vol. 10, No. 3, pp. 291-309, 1988.

[9] R. V. Mayorga, F. Janabi-Sharifi and A. K. C. Wong, "A Fast Approach for the Robust Trajectory Planning of Redundant Robot Manipulators", *Journal of Robotic Systems*, Vol. 12, No. 2, 1995.

[10] K. D. Rueb and A. K. C. Wong, "Knowledge-based Visual Part Identification and Location in a Robot Workcell", *Int. J. Mach. Tools Manufact.*, Vol. 28, No. 3, pp 235-249, 1988.

[11] A. K. C. Wong, G. R. Heppler, D. N .C. Tse and K. D. Rueb, "Robotic Vision Technology for Space Station and Satellite Applications", *Acta Astronauticas Journal*, Vol. 29, No. 12, pp. 911-930, 1993.

[12] A. K. C. Wong, "3D Vision and Modeling", *Proc. of 2nd Int'l Conf. on Mechatronics and Machine Vision in Practice*, Hong Kong, pp. 39-48, Sept. 12-14, 1995.

[13] A. K. C. Wong, P. Yu and X. Liang, "Elliptical Curve Detection Through Curve Data Grouping and Fitting", to be appeared on *CVGIP*, 1997.

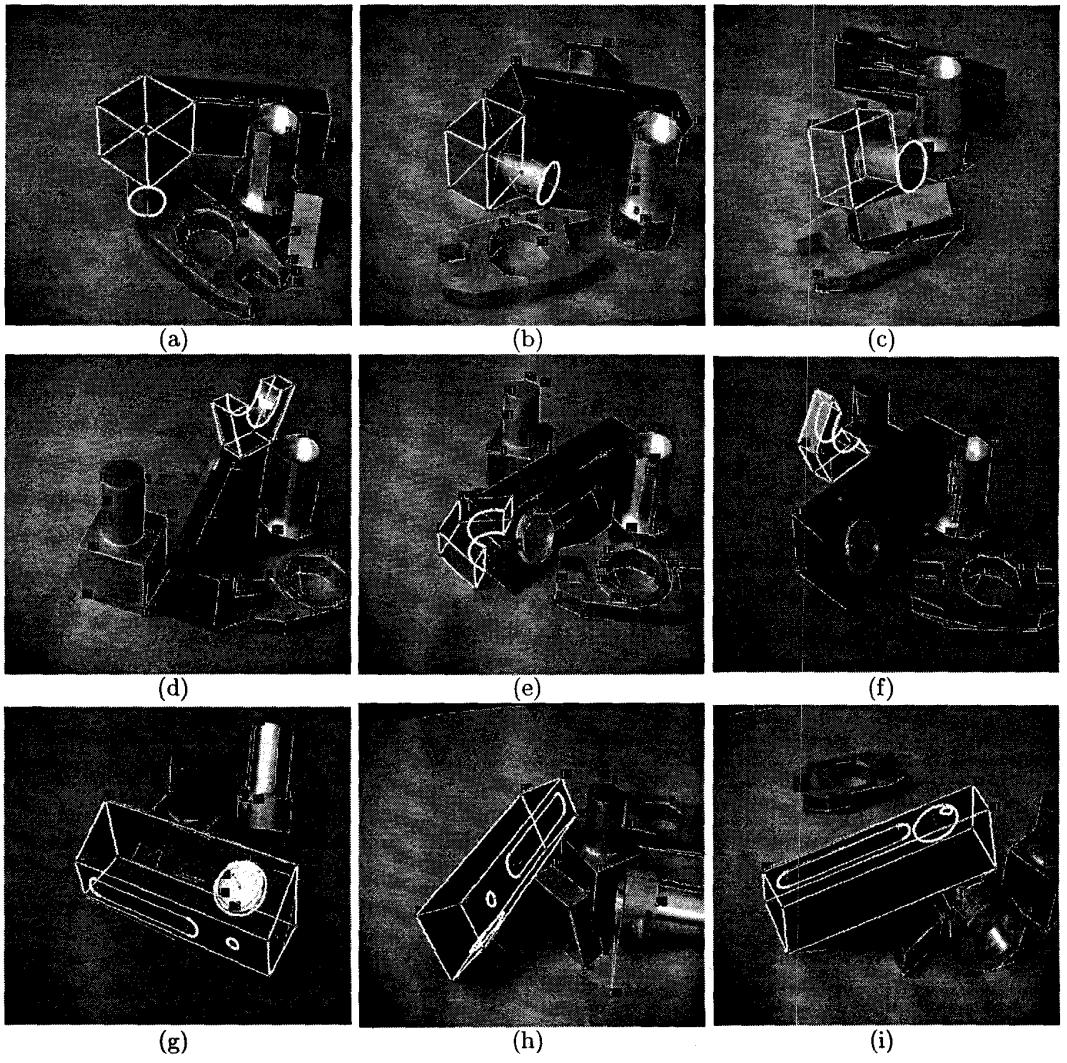


Figure 8: Sample recognition results for the three objects posed at random orientations: (a)-(c) the recognition of the grenade; (d)-(f) the recognition of the bridge; (g)-(i) the recognition of the pen-holder.