COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 50, 177–187 (1990)

NOTE

3D Curved Object Recognition from Multiple 2D Camera Views*

CHENG-HSIUNG LIU

Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China

AND

WEN-HSIANG TSAI†

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China

Received November 29, 1988; accepted May 24, 1989

A new approach to 3D object recognition using multiple 2D camera views is proposed. The recognition system includes a turntable, a top camera, and a lateral camera. Objects are placed on the turntable for translation and rotation in the recognition process. 3D object recognition is accomplished by matching sequentially input 2D silhouette shape features against those of model shapes taken from a set of fixed camera views. This is made possible through the use of top-view shape centroids and principal axes for shape registration, as well as the use of a decision tree for feature comparison. The process is simple and efficient, involving no complicated 3D surface data computation and 3D object representation. The learning process can also be performed automatically. Good experimental results and fast recognition speed prove the feasibility of the proposed approach.

I. INTRODUCTION

In industrial automation tasks such as machine parts sorting and assembly, it is often found necessary to recognize 3D objects. Besl and Jain [1], and Chin and Dyer [2] include extensive surveys on 3D object recognition research. In this paper, a new method for 3D object recognition using multiple 2D object silhouette shapes taken from distinct camera views is proposed.

Most existing methods [3–6] try to acquire 3D object surface data and transform them into certain object representations for use in the recognition process. This often involves the difficult problems of system calibration, surface data computation, and object modeling. Such object recognition methods may be said to be based on the principle of *3D recognition by 3D shape analysis*. If the aim of recognition is just to classify the object, it might be desirable to avoid direct 3D shape analysis. The idea of *3D recognition by 2D shape analysis* has been employed in several investigations [7–13]. The proposed method falls into this category. An advantage of this approach is that the well-developed 2D shape analysis techniques can be

*This work was supported financially by ERSO, ITRI under Grant MIST-E7502.

†To whom all correspondence should be sent.

0734-189X/90 $3.00

utilized. In the following, existing methods based on the principle of 3D recognition by 2D analysis are briefly reviewed.

Wallace and Wintz [7] used the Fourier descriptors of 2D silhouette shapes to recognize 3D aircrafts. A library of 2D shape descriptors for all discrete viewing directions covering the entire spherical solid angle is created. Recognition is accomplished by matching input shape descriptors against all the data in the library. Similar techniques were also used by Dudani *et al.* [8].

Watson and Shapiro [9] matched 2D perspective views of 3D objects with object models consisting of closed connected curved edges of the objects. Input 2D scenes are processed into curves which are also described by Fourier descriptors. Object recognition is accomplished by comparing the 2D perspective projections of the model curve with the input curve after the former is properly rotated and translated.

Wang *et al.* [10] also recognized 3D objects by 2D silhouette shapes. Each object model consists of the three principal axes, the principal moments, and the Fourier boundary shape descriptors of the silhouette shapes as viewed from the three principal axes. To recognize an input object, at least three silhouette views from distinct directions have to be taken. Silhouette boundaries are then combined to produce an object from which the moments and the Fourier descriptors can be computed and matched against the model library.

Goad [11] used multiple-view object models, each consisting of 218 different 3D views. Line segments or edges are used as the shape feature. The recognition process involves backtracking search for matches between input and model object edges.

Chakravarty and Freeman [12] set up multiview object models using the so-called characteristic views, each representing a set of perspective projections of a given object with an identical topological property. Matching is performed by applying line-junction labeling constraints on each edge found in input object views.

Silberberg *et al.* [13] used the general Hough transform technique to match input 2D line segments and edge junctions with 3D model line segments and vertices. For each pair of line segments being matched, the model line is projected onto the image line, incrementing the corresponding cell in the Hough accumulator array if the matching is successful.

In all the above approaches, it is not necessary to compute 3D object surface data. Only 2D images are processed to extract relevant object features for use in the recognition process. However, some of the approaches require 3D object representations to establish object models for which computer graphics is often used as the tool. The others use features extracted from 2D object shapes directly as the object model. An advantage of the latter approach is the ease of model establishment in the learning phase. The method proposed in this paper belongs to this type of approach. The features used in the method are simple geometric properties and moment invariants extracted from 2D silhouette shapes. And object models are organized in terms of a decision tree. Matching of input object shape features against the object models is accomplished by traversing the decision tree until a tree leaf is reached. The use of decision trees for recognition greatly improves the recognition speed.

Automatic learning of reference objects is always desired in an object recognition system. An automatic decision tree construction algorithm is also proposed in this paper. This facilitates model updating when objects to be recognized are added or deleted.

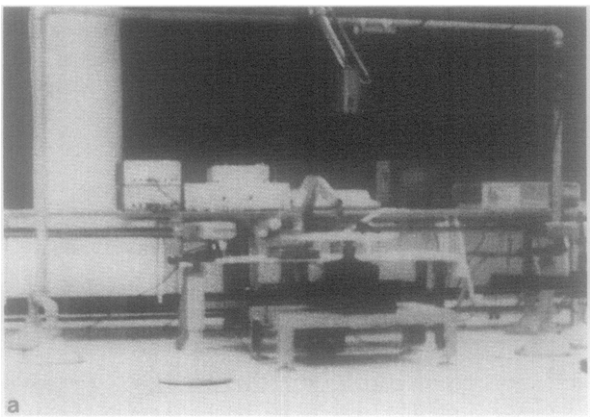
In the remainder of this paper, an overview on the proposed 3D object recognition system is first presented. Detailed discussions on the object learning and recognition procedures follow next. The experimental results are presented finally.

II. SYSTEM OVERVIEW

A. *Basic Idea of 3D Recognition by 2D Shape Analysis*

The proposed 3D object recognition system is designed mainly for use in industrial automation. Industrial parts to be recognized are placed on the turntable with a robot arm. Every stable state of each given industrial part is considered as a distinct object in the recognition process. Object images are taken by the TV cameras and thresholded into binary silhouette shapes for further feature extraction. A prototype system constructed for this study is shown in Fig. 1a and the system configuration is illustrated in Fig. 1b.

To recognize an unknown object which is placed on the turntable, the system first takes a top view of the object, using the top camera, and then *normalizes* the top-view shape by translating the shape centroid to right under the top camera and rotating the principal axis of the shape to align with the X-axis of the image plane.



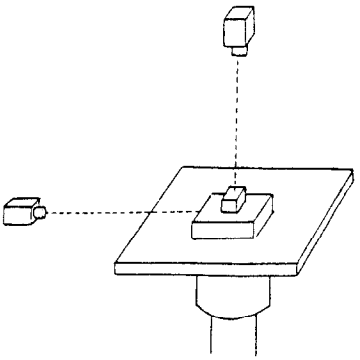


FIG. 1. The proposed 3D object recognition system. (a) A prototype system constructed for experiment. (b) System configuration.

This makes the object shape always appear to be identical in position and orientation. Another 2D object image is taken again and matched against the object models (organized in the form of a decision tree). The details of matching will be described later. If the top-view shape is already discriminable, the recognition is completed. This speeds up the recognition process in general because a lot of industrial parts are 2D in nature in most of their stable states and can be easily discriminated from one another by their top views.

If the top-view shape is inadequate for discrimination (i.e., if several object models have the same top-view shape as that of the object being recognized), the lateral camera is activated to take the side-view image of the object from a fixed direction with respect to the centroid and the principal axis of the top-view shape. Since only one lateral camera is used, image taking from various directions is made possible by rotating the object using the turntable. The side-view object shape is then matched against the object models for further discrimination. If the shape is uniquely discriminable from those of other objects, the process is terminated. Otherwise, a distinct side-view shape of the object is taken from another lateral direction, and the above process is repeated, until the object is discriminable from a certain lateral direction. As a summary, the above idea of hierarchical or multi-stage recognition is illustrated in Fig. 2.

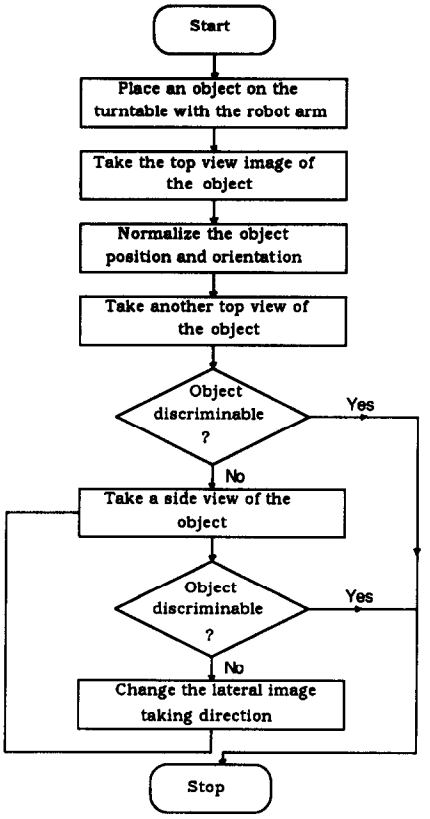


FIG. 2. Object recognition process.

B. *Shape Features for Object Recognition*

Top-view or side-view shape recognition mentioned previously is accomplished by shape feature matching. Six types of feature are used in this study. Three of them are moment invariants and the others are geometric properties. Moment functions are also used to compute shape centroids and principal axes.

Let B denote a set of N black points in a binary silhouette shape. Each point in B is associated with value 1. The coordinates of the ith point in B are (x_i, y_i). The (p, q) moment of B is defined as $m_{pq} = \sum_{i=1}^{N} x_i^p y_i^q$ and the (p, q) central moment of B is defined as $M_{pq} = \sum_{i=1}^{N} (x_i - \bar{x})^p (y_i - \bar{y})^q$, where $\bar{x} = m_{10}/m_{00}$, $\bar{y} = m_{01}/m_{00}$, and $(\bar{x}, \bar{y})$ is the centroid of B. The principal axis of shape B is defined as the line L about which the moment of inertia of B is minimum. It can be shown [14] that L goes through the centroid of B, and that the slope $\tan\theta$ of L satisfies the equation $\tan 2\theta = 2M_{11}/(M_{20} - M_{02})$.

The six features used for recognition are as follows:

(1) $F_1 = m_{00} = M_{00}$ which is the area of B (since B is the silhouette shape boundary, F_1 actually is the perimeter of the boundary);

(2) $F_2 = [(M_{02} - M_{20})^2 + 4M_{11}^2]/M_{00}$ which is the eccentricity of the object;

(3) $F_3 = (M_{20} + M_{02})/M_{00}$ which is the moment of inertia around the centroid;

(4) F_4 = the number of black pixels along the principal axis inside the shape boundary;

(5) F_5 = the vertical extent of the minimum-sized rectangle circumscribing the object shape;

(6) F_6 = the horizontal extent of the minimum-sized rectangle circumscribing the object shape.

The above six features are extracted from each view of the object and form a feature vector F, i.e., $F = (F_1, F_2, \ldots, F_6)$. Among the six features, F_1 can be used to discriminate objects of different perimeters. F_2, F_3, and F_4 are useful object shape properties. F_5 and F_6 can be used to measure object sizes. The features were experimentally selected and were found adequate to discriminate the set of artificial objects made for this study (see Section V for details). Additional types of features may be necessary if more complicated objects are to be recognized although the six features types are believed to be sufficient for most industrial objects.

C. *Image Processing*

Preprocessing of object images is necessary before feature extraction and matching. The first step is bilevel thresholding to obtain the binary silhouettes of object shapes. The moment-preserving thresholding method proposed by Tsai [15] is used. Since it is assumed that the recognition system can be set up in a controllable environment, the lighting condition can be adjusted to get better contrast between an object and the background, ensuring successful segmentation of the object shape out of the background.

The next task of image processing is object boundary tracing. For this, the binary silhouette image is scanned from the top to find a point on the object boundary, which is then used as the starting point of a simple boundary tracing algorithm [14].

At the end of boundary tracing, the previously described features can then be extracted for shape matching.

III. LEARNING OF OBJECT MODELS USING A DECISION TREE

Object models are organized in the form of a hierarchical decision tree. This is done in the learning phase. Decision tree construction is described in detail in this section, including feature extraction, feature clustering, and tree setup. It is emphasized that these steps can all be performed automatically without human interaction.

A. *Feature Extraction*

With the feature types as defined in Section II.B, the following is the procedure to extract relevant features from each reference object placed on the turntable in a certain stable state.

ALGORITHM 1. Feature extraction from a reference object shape.

Step 1. Use the top camera to take the top-view image of the object. Threshold the image into a binary silhouette shape and compute the centroid and the principal axis of the shape.

Step 2. Control the turntable to normalize the object position and orientation as described previously.

Step 3. Take another top-view image of the normalized object, threshold the image, trace the thresholded object shape boundary, and compute the feature values.

Step 4. For each of the lateral angles $\theta = 0°, 45°, 90°, 135°, \ldots, 315°$, perform the following two steps.

4.1. Control the turntable to translate and rotate the object so that the optical axis of the lateral camera goes through the top-view shape centroid from the specified lateral direction θ.

4.2. Take the side-view image of the object, threshold the image, trace the object shape boundary, and compute the feature values.

The purpose of normalizing the object in Step 2 before feature computation in Step 3 is to reduce perspective transformation effect on the top-view object shape.

B. *Feature Clustering*

For feature comparison, a mismatch measure has to be defined. Let $F_1 = (F_{11}, F_{12}, \ldots, F_{16})$ and $F_2 = (F_{21}, F_{22}, \ldots, F_{26})$ be two feature vectors to be compared. The mismatch measure $d(F_1, F_2)$ used in this study is defined as

$$d(F_1, F_2) = \sum_{j=1}^{6} |F_{1j} - F_{2j}| / (1 + |F_{1j}| + |F_{2j}|), \tag{1}$$

where the denominator is used for feature magnitude normalization because the six feature types are different in magnitude. The value "1" is included to avoid the occurrence of zero denominator values. Note that the measure function is symmetric, i.e., $d(F_1, F_2) = d(F_2, F_1)$, and that $d(F_1, F_2) = 0$ when $F_1 = F_2$.

Two feature vectors are said to be *different* (or *similar*) if the mismatch measure between them is larger than (or not larger than) a preselected threshold value. Two

object shapes are said to be different (or similar) if their feature vectors are compared to be different (or similar). Two distinct threshold values are used for such similarity comparison, one for top-view features and the other for side-view features. The reason is that the two cameras used for top-view and side-view image taking are different in nature and in distance from the object. To choose either of the two threshold values, let $G_1, G_2, \ldots, G_n$ be n groups of similar top-view (or side-view) object shapes with their similarity being visually determined, and let

$$d_i = \max_{j,k} d\left(F_j^i, F_k^i\right),$$

where F_j^i and F_k^i are the feature vectors of any two shapes in group G_i. That is, let d_i be the maximum feature distance in G_i. Then the desired threshold value is chosen to be $d = \sum_{i=1}^{n} d_i/n$. Note that the use of such a threshold value in the feature clustering algorithm described next might result in a set of object groups slightly different from the visually-determined groups G_1 through G_n. But the resulting object groups are experimentally found good enough for object discrimination in the recognition phase.

Given a set of m feature vectors, $G = \{F_1, F_2, \ldots, F_m\}$, where $F_i = (F_{i1}, F_{i2}, \ldots, F_{i6})$, $1 \le i \le m$, the procedure to cluster the feature vectors is as follows. The effect of feature clustering is to partition the set of objects into groups, each group including all the objects with similar shape features.

ALGORITHM 2. Feature clustering.

Step 1. Set index $j = 1$ initially.

Step 2. Select arbitrarily a feature vector F_k from G. Form a new cluster G_j as $\{F_k\}$, and let the cluster center C_j of G_j be F_k. Remove F_k from G.

Step 3. For each F_i in G, if F_i and C_j are similar then merge F_i into cluster G_j. Remove F_i from G, and update the center C_j of G_j as the average of all feature vectors contained in G_j currently.

Step 4. If G is empty, then exit; else set $j = j + 1$ and go to Step 2 to create another cluster.

The above algorithm is simple and deterministic, compared with other interactive clustering algorithms. After the algorithm is performed, each cluster will include a center which is then adopted as the *representative feature vector* of the cluster for use in the decision tree, as described next.

C. *Decision Tree Setup*

The use of a decision tree facilitates hierarchical decision making using the feature vectors, avoiding exhaustive feature matching and speeding up the recognition process.

The root node of the decision tree can be considered to include all the objects in a single cluster. Since most industrial parts are flat (i.e., close to 2D in shape), they may be discriminated more easily from their top-view shapes. Therefore, it is justified to use top-view object shape features first for object recognition. Accordingly, the child nodes of the tree root are generated according to the top-view shape feature vectors. More specifically, we use Algorithm 2 to cluster the feature vectors of the top-view shapes of all the objects. Each resulting cluster, which includes one

or more objects, forms a child node of the tree root. They are called *the first-level nodes* of the tree.

If any first-level node includes just a single object, it is called *a terminal node*, which means that the object can be fully discriminated from all other objects using just the top-view shape features. On the contrary, if a first-level node includes several objects, their side-view shape features must be used for further discrimination. A question here is which lateral direction should be chosen first (i.e., which feature vector among those of the lateral directions, $0°, 45°, 90°, 135°, \ldots, 315°$, should be used first in discriminating the objects included in the current first-level node). For this, a simple feasible criterion is to choose the lateral direction whose feature vector can be used to "decompose" (using the clustering algorithm, Algorithm 2) the current node into the largest number of subclusters. Based on this criterion, each nonterminal tree node can be decomposed recursively into terminal nodes, possibly after traversing several tree levels. A detailed tree setup algorithm is described in the following.

ALGORITHM 3. Decision tree setup.

Step 1. Create the root node of the tree which includes all the reference objects.

Step 2. Apply Algorithm 2 to the set of the top-view shape feature vectors of all the objects to generate the first-level child nodes, each node corresponding to a cluster output by Algorithm 2.

Step 3. For each of the nonterminal node N, perform the following steps recursively until no new nonterminal node is generated.

3.1. For each lateral direction θ of the eight possible ones ($0°, 45°, 90°, \ldots, 315°$), apply Algorithm 2 to the set of the shape feature vectors of direction θ of all the objects included in node N to generate a group of clusters. Denote the number of the clusters as $NO(\theta)$.

3.2. Find the direction θ_m such that $NO(\theta_m) = \max_\theta NO(\theta)$.

3.3. Create the next-level child nodes of N to be the clusters of direction θ_m generated in Step 3.1.

The direction θ_m found in Step 3.1 will be called *the most effective view* for clustering the objects included in nonterminal node N. For the root node, the most effective view is just the top view of the object. The most effective views will be used in the recognition process.

IV. INPUT OBJECT RECOGNITION

Input object recognition may be regarded as a traversal process in the decision tree from the tree root to a terminal node. The recognition process can be described as an algorithm as follows.

ALGORITHM 4. Object recognition process.

Step 1. Place the object to be recognized on the turntable. Set the tree root as the current node N.

Step 2. Take the top-view image of the object using the top camera.

Step 3. Threshold the image and normalize the object position and orientation according to the centroid and the principal axis of the resulting binary silhouette object shape.

Step 4. Take another top-view image of the normalized object, threshold the image, and trace the object boundary.

Step 5. Extract feature vector F from the result of the last step using Algorithm 1.

Step 6. Compare F with the representative feature vector of each child node of the current node N. If F is different from all the representative feature vectors, then the input object is rejected as an unknown object. Otherwise, let node N' be the one with its feature vector similar to F.

Step 7. If N' is a terminal node, then the object is recognized as the one included in N' and the recognition process is terminated. Otherwise, continue.

Step 8. Find out the most effective view θ associated with N' from the decision tree. Activate the turntable to rotate the object and use the lateral camera to take a side-view image of the object from direction θ. Threshold the image and trace the object boundary.

Step 9. Set N' as the current node N and go to Step 5.

V. EXPERIMENTS AND DISCUSSIONS

A. *Experimental Results*

Ten artificial machine parts as shown in Fig. 3 are used for experiment, which can be placed in 35 distinct stable states. Execution of the learning process resulted in the 3-level decision tree as shown in Fig. 4. Note that a nonterminal node needs no further splitting if it includes more than one stable state all of which belong to a single machine part. Two such nodes can be found at the bottom level of the tree (one including states 23, 24, and 25, and the other including states 30 and 31). In order to verify that the decision tree we constructed can be used to recognize the objects correctly no matter how many stable states were involved in the decision tree, four sessions of recognition experiments have been done, each including the recognition of more than 20 stable states. The results are summarized in Table 1. The average recognition rate is about 98%. Recognition of each part takes no more than 5 s. The program was written in the C language. Since the artificially-made objects are so close to real ones, expected performance of the proposed approach on actual machine parts will be reasonably close to the foregoing experimental results.

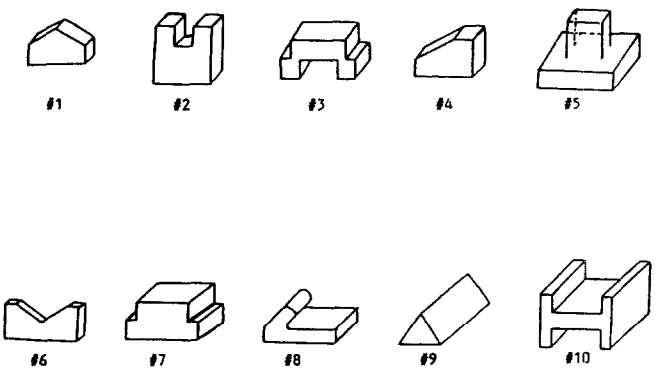


FIG. 3. The ten artificial machine parts used in the experiments (only one stable state of each part is shown).

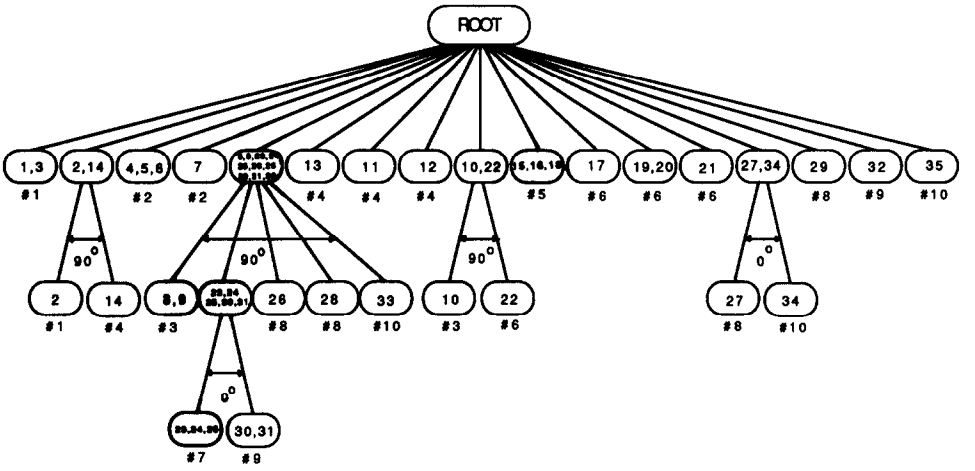


FIG. 4. The decision tree for the ten machine parts shown in Fig. 3 (the numbers in the nodes specify stable state numbers, those outside the nodes specify machine part numbers, and the degree numbers specify the most effective views).

B. *Suggestions for Possible Improvements*

The recognition speed seems a little slow, but it was the result of implementing the proposed algorithms on an IBM PC/XT compatible microcomputer. The average recognition time for each part could be greatly reduced to meet practical applications if an IBM PC/AT microcomputer coupled with a 80287 numerical processor can be used.

The recognition errors have been analyzed. One source of error is improper threshold selection for feature comparison, resulting in assignment of input objects to incorrect tree nodes. Another source causing the errors is the camera sensitivity to light changes in the environment, resulting in the undesirable changes of shape boundaries and object features. A possible improvement is to use autoregressive models to represent object boundaries, making extracted features less sensitive to boundary distortion. Another improvement is to adopt more sophisticated pattern classification algorithms to discriminate object shapes instead of using the simple deterministic mismatch measure described by Eq. (1). This will reduce the effect of feature distortion due to nonperfect boundary extraction.

TABLE 1
The Results of Four Sessions of Recognition Experiments

Session	Number of stable states recognized in the session	Number of stable states correctly recognized	Recognition rate (%)
1	35	35	100
2	28	27	96
3	24	23	95
4	20	20	100

VI. CONCLUSIONS

A new approach to recognizing 3D curved objects by 2D silhouette shape analysis is proposed. The approach adopts simple shape features and decision trees to accomplish the recognition work. The top views of object shapes are used first for object discrimination, followed by the use of object side views. Objects are viewed from fixed lateral directions after object position and orientation normalization using top-view shape centroids and principal axes. The computation is not complicated, and the speed is reasonable. The learning process for decision tree setup can be automated. The high recognition rate proves the feasibility of the proposed approach.

REFERENCES

1. J. Besl and R. C. Jain, Three-dimensional objection recognition, *Comput. Surveys* **17**, No. 1, 1985.
2. R. T. Chin and C. R. Dyer, Model-based recognition in robot vision, *Comput. Surveys* **18**, No. 1, 1986.
3. R. Nevatia and T. O. Binford, Description and recognition of curved objects, *Artif. Intell.* **8**, No. 1, 1977.
4. M. Oshima and Y. Shirai, Object recognition using three-dimensional information, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 4, 1983.
5. Y. Sato and I. Honda, Pseudo-distance measures for recognition of curved objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 4, 1983.
6. W. N. Martin and J. K. Aggarwal, Volumetric descriptions of object from multiple-views, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 2, 1983.
7. T. P. Wallace and P. A. Wintz, An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors, *Comput. Vision Graphics Image Process.* **13**, 1980, 96–126.
8. S. A. Dudani, K. J. Breeding, and R. B. McGhee, Aircraft identification by moment invariants, *IEEE Trans. Comput.* **C-26**, No. 1, 1977.
9. L. T. Watson and L. G. Shapiro, Identification of space curves from two-dimensional perspective views, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-4**, No. 5, 1982.
10. Y. F. Wang, M. J. Maggee, and J. K. Aggarwal, Matching three-dimensional objects using silhouettes, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 4, 1984.
11. C. Goad, Special-purpose automatic programming for 3D model-based vision, in *Proceedings, Image Understanding Workshop Arlington, VA June 1983*, pp. 94–104, Science Applications, Arlington, VA, 1983.
12. I. Chakravarty and H. Freeman, Characteristic views as basis for three-dimensional object recognition, in *Proceedings, SPIE Conference on Robot Vision, Arlington, VA, May 1982*, Vol. 336, pp. 37–45, SPIE, Bellingham, WA, 1982.
13. T. M. Silberberg, L. S. Davis, and D. Harwood, "An iterative Hough procedure for three-dimensional object recognition," *Pattern Recognit.* **17**, No. 6, 1984.
14. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. II, Academic Press, New York/San Francisco/London, 1980.
15. W. H. Tsai, Moment-preserving thresholding: A new approach, *Comput. Vision Graphics Image Process.* **29**, 1985, 377–393.