Third Int. Conf. Industrial & Eng. Applications of AI and Expert Systems, Charleston, S.C. July 90

Recognizing Objects in Range Images and Finding Their Position in Space

Jun Ohya*1, Daniel DeMenthon*2, and Larry S. Davis*2

*1 Nippon Telegraph & Telephone Corp.

1-2356 Take, Yokosuka-shi, Kanagawa, 238-03, Japan

*2 University of Maryland, College Park, MD 20742

Abstract

We present a method for recognizing polyhedral objects from range images. An object is said to be recognized as one of the models of a library of object models when many features of the model can be made to match the features of the observed object by the same rotation-translation transformation (the *object pose*). In the proposed approach, the number of considered pairs of image and model features is reduced by selecting at random only a few of all the possible image features and matching them to appropriate model features. The rotation and translation required for each match are computed, and a robust LMS (Least Median of Squares) method is applied to determine clusters in translation and rotation spaces. The validity of the object pose suggested by the clusters is verified by a similarity measure which evaluates how well a model in the suggested pose would fit the original range image. The pose estimation and verification are performed for all models in the model library. The recognized model is the model which yields the smallest value of the similarity measure, and the pose of the object is found in the process.

1 Introduction

Recognizing three-dimensional (3-D) objects in range images and finding the poses of these objects are major topics in computer vision [1,2,3,4,5,6]. This paper addresses these problems for polyhedral objects with six degrees of freedom.

The proposed method is similar to a generalized Hough transform[7]. Hough transforms are attractive because of insensitivity to noise and robustness in presence of occlusions.

However, in Hough transforms, choosing an optimal size for the bins of the accumulator arrays of the parameter spaces is a difficult problem. For this reason, we propose to detect clusters in parameter space by a robust clustering technique[8].

Another originality of the proposed method resides in the fact that the number of matches between image and model features is reduced by selecting at random only a few of all the possible image features, and matching them to appropriate model features. The method considers combinations of range image features which generate unique pose estimates. Clustering is performed separately for rotation and translation. Points in the rotation parameter space are determined by matching pairs of surface normals from the range image (obtained by randomly selecting pairs of pixels and their neighbors) with pairs of normals from the faces of the models. Similarly, points in the 3-D translation parameter space are generated by matching triples of surface patches obtained from range pixels and their neighbors (chosen at random in the range image) with triples of planar faces of the models.

A LMS (Least Median of Squares) method[8] is applied to cluster detection in these parameter spaces. The LMS method can accurately estimate centers of clusters from point data in a multi-dimensional space in the presence of outliers. In each parameter space, spheres are considered with their centers at nodes of a 3-D grid and the LMS method is applied inside each sphere. The spheres are then recentered at the found clusters, and the process is repeated. After all spheres have settled, the spheres which contain the largest number of points are considered to be centered at the peaks of the clusters.

Finally, an hypothesis verification method is proposed. The cluster detection method generates hy-

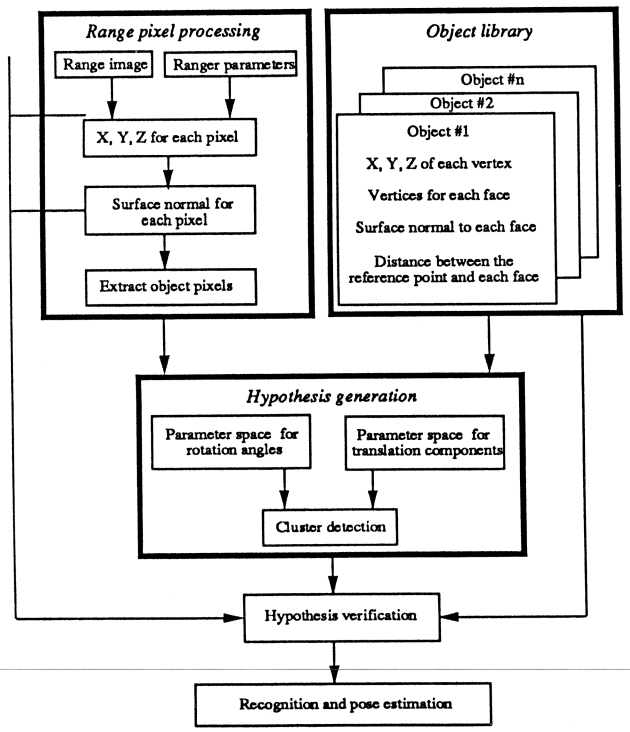

Figure 1: Flow chart of the proposed method

potheses for the six pose parameters. The generated hypotheses are verified by a similarity measure which evaluates how well a model of a given pose fits the original range image. The hypothesis generation and verification are performed for all models in a model library. The recognized model corresponds to the smallest value of the similarity measure, and the pose parameters of the object in the range image are obtained in the process.

The different steps required by the method are (1) Object modeling, (2) Processing of range image pixels, (3) Hypothesis generation (pose estimation), and (4) Hypothesis verification. A flow chart of the method is given on Figure 1. In the following sections, each of these steps is analyzed, and experimental results are presented.

2 Object Modeling

The polyhedral objects to be recognized need to be modeled in advance and stored in a model library. The object models are used in the hypothesis generation and in the hypothesis verification. Model descriptions are specified in a coordinate system linked to the models and centered at a specific point of the model called the *reference point.* These descriptions include (1) The vertex coordinates, (2) The vertices of each face, (3) The surface normal to each face, and (4) The distance between each face and the reference point.

3 Processing of Range Image Pixels

Using the known parameters of the range scanner, it is straightforward to obtain cartesian coordinates of the world points corresponding to pixels in the range image. Since this transformation is specific to the type of range scanner used, we will not give further details.

The proposed method also requires obtaining the surface normals to the surface patches surrounding world points. For a pixel and its corresponding world point, we consider the eight neighbor pixels and the eight corresponding world points. The surface normal to the patch of these world points is obtained by averaging six normalized cross-products of the vectors which join these world points.

4 Hypothesis Generation (Pose Estimation)

The pose of an object can be defined by a rotation matrix $\mathbf{R}$, whose components are trigonometric functions of the rotation angles θ, ϕ, ψ around the three world coordinate axes, and by a three component translation vector $\mathbf{T}$.

4.1 Rotation parameters

In our implementation, a number of pairs of surface normals are computed from pairs of range image pixels (their neighbors are also needed to determine the normals) chosen *at random* in the image. Each pair of surface normals is matched to all pairs of surface normals from the current model provided they make similar angles.

In order to match one surface normal $\mathbf{n}_1$ from a model with one surface normal $\mathbf{n'}_1$ obtained from the range image, we can use any rotation whose axis is parallel to the plane which is normal to the vector $\mathbf{n'}_1 - \mathbf{n}_1$. Now, in order to match a *pair* of surface normals $\mathbf{n}_1$ and $\mathbf{n}_2$ from a model with a pair of surface normals $\mathbf{n'}_1$ and $\mathbf{n'}_2$ obtained from the range image and making similar angles, we have to use a rotation whose axis is parallel both to the plane perpendicular to $\mathbf{n'}_1 - \mathbf{n}_1$ and to the plane perpendicular to $\mathbf{n'}_2 - \mathbf{n}_2$. In other words, the

direction $\mathbf{k}$ of the axis for this rotation is given by the cross product between $\mathbf{n'}_1 - \mathbf{n}_1$ and $\mathbf{n'}_2 - \mathbf{n}_2$. Next we find the angle α of rotation. This angle α is the angle between the plane parallel to $\mathbf{k}$ and $\mathbf{n}_1$ and the plane parallel to $\mathbf{k}$ and $\mathbf{n}_2$, ie. the angle between the normals to these planes. Thus we compute the cross-products between $\mathbf{k}$ and $\mathbf{n}_1$ and between $\mathbf{k}$ and $\mathbf{n}_2$, and find the angle between these two vectors.

We then obtain the rotation matrix $\mathbf{R}$ given the rotation axis $\mathbf{k} = (k_1, k_2, k_3)$ and the angle α. The rotation matrix is given by the Rodrigues formula

$$\mathbf{R} = \mathbf{I}_3 + \sin\alpha\mathbf{U} + (1 - \cos\alpha)\mathbf{U}^2 \qquad (1)$$

where $\mathbf{I}_3$ is the 3×3 identity matrix and $\mathbf{U}$ is the skew-symmetric matrix of the coordinates of $\mathbf{k}$

$$\mathbf{U} = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}$$

Once the rotation matrix is obtained, finding the three rotation angles θ, ϕ, ψ is straightforward. See for example [6] for details.

Therefore for each pair of normal vector matches, a point in the $\theta - \phi - \psi$ space is obtained. When the number of these points is sufficiently large, the process is stopped. The clusters in the rotation space are detected by the LMS method detailed in a subsequent section.

4.2 Translation Parameters

For a pixel in the range image and its neighbors the world point P and the local unit surface normal $\mathbf{n}$ can be computed. They define a planar patch. If a planar face of a model at a distance m from the model reference point is matched to this world patch, the matching constrains the reference point to be translated by a vector $\mathbf{T}$ from the origin of the world coordinate system onto a plane parallel to the chosen planar face at a distance m. In other words the translation is such that

$$(\mathbf{T} - \mathbf{P}) \cdot \mathbf{n} = m \qquad (2)$$

where the "$\cdot$" symbol expresses the dot product between vectors. This is the equation of the plane which is the locus of the positions that the reference point of the model can occupy. If now three non coplanar surface patches computed from three pixels and their neighbors are matched to three faces of a model, the reference point is constrained to be at the intersection of three planes. Each of the three planes has an equation of the type of Equation 2, where P and $\mathbf{n}$ are the world point and the unit normal for each surface patch. The position of the model reference point after the match is the translation vector $\mathbf{T} = (T_X, T_Y, T_Z)$ found by solving the system of three equations. In our implementation, triples of pixels and their neighbors are chosen *at random* in the range image; if the corresponding surface patches are found to be non coplanar, these triples of planar patches are matched to triples of faces for all the models of the library, provided their surface normals make comparable angles. Each new combination of matches of triples of pixels and triples of faces generates a new point (T_X, T_Y, T_Z) in the translation space. When the number of these points is sufficiently large, the process is stopped. Then the LMS method is used for cluster detection.

4.3 Cluster Detection by the LMS method

The LMS method has been applied successfully to the task of finding centers of clusters in q-dimensional spaces[8,9]. If the number of clusters K is known, and if the number of inliers is more than 50% of the total number of data points, then the LMS method can accurately detect cluster centers[8].

However in parameter spaces the number K of clusters is unknown because many spurious clusters may occur. Besides, the number of data points in the cluster corresponding to the actual object pose is in general less than half the total number of data points. For these reasons the LMS method should be applied to small volumes rather than to the entire parameter space.

A 3-D grid is considered in the parameter space (Figure 2). Spheres are centered at the nodes of the 3-D grid. The radius h of the spheres is chosen equal to the mesh size of the grid so that the spheres overlap and cover the entire parameter space. Within each sphere the LMS method is applied and calculates the center of a cluster. Then the center of each sphere is moved to the cluster center found inside. The LMS method then calculates a new cluster center within each sphere in its new position. This process is repeated until no more significant sphere displacement is required.

Finally, the positions of the cluster centers found in the spheres which contain the largest amounts of points are retained as candidates for representing the actual object pose. Several of the largest clusters are kept because the actual pose may correspond to the second or third largest cluster, or several object poses may be equally possible because of object symmetries. The final verdict on these candidates is left to the verification stage.

We now focus on the cluster detection within each sphere. A new cluster center is calculated by applying

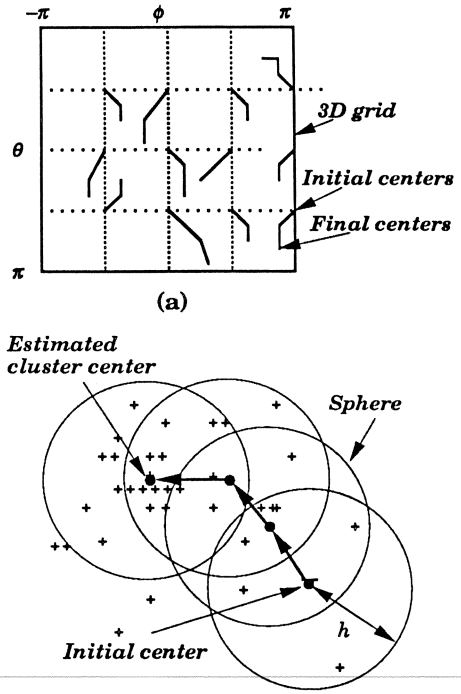

Figure 2: Cluster detection by a LMS method.
(a) Initial and final positions of sphere centers on a 3-D grid.
(b) Progress of estimations of sphere centers

the 1-D LMS method independently for each of the three coordinate of the n points contained in a sphere. Let s_i $(s_1 \leq s_2 \leq \cdots \leq s_n)$ be a sorted list of coordinates; let M and R^2 denote the value minimizing the median of residual squares and its corresponding median of residual squares, respectively. Then,

$$M = (s_{r+k} + s_r)/2 \quad (3)$$

$$R = (s_{r+k} - s_r)/2 \quad (4)$$

where $(s_{r+k} - s_r) = \min(s_{i+k} - s_i)$, $i \leq n - k$.
In these equations, k is set to $n/2$ in order to avoid local maxima. The standard deviation σ is estimated by

$$\sigma = 1.4826 \times (1.0 + \frac{5.0}{n-1}) \times R \quad (5)$$

Then, the coordinate X_c of the center of the cluster is

$$X_c = \frac{\sum_{i=1}^{n} W_X(X_i) X_i}{\sum_{i=1}^{n} W_X(X_i)} \quad (6)$$

where X_i is the coordinate of point i, and W_X is a weight function defined by

$$W_X = \begin{cases} 1 & \text{if } \frac{\Delta X_i}{\sigma} \leq 2.0 \\ 0 & \text{if } \frac{\Delta X_i}{\sigma} \geq 3.0 \\ 3 - \frac{\Delta X_i}{\sigma} & \text{otherwise} \end{cases} \quad (7)$$

where ΔX_i is the distance between M_X and X_i; M_X and σ_X are M and σ for the X axis, respectively. The other coordinates for a cluster center are obtained the same way.

5 Hypothesis Verification

The candidates detected in Section 4.3 need to be verified so that the actual poses can be obtained. A model is projected onto the original range image using the range scanner parameters and the estimated pose parameters. Then, a similarity measure described below is computed.

The basic idea of the similarity measure is to examine how well visible faces of a projected model fit to the original range image. For the pixels included in the projected visible faces, measures of differences in ranges and surface normals between the range image and the projected model faces are calculated. The similarity measure S is defined by

$$S = \frac{W_1}{NJ} \sum_{j=1}^{NJ} \frac{1}{NPX_j} \sum_{i=1}^{NPX_j} (\Delta surfn)_i + \frac{W_2}{NJ} \sum_{j=1}^{NJ} \frac{1}{NPX_j} \sum_{i=1}^{NPX_j} (\Delta xyz)_i \quad (8)$$

where NJ is the number of visible faces, NPX_j is the number of pixels on the visible face j, $(\Delta surfn)_i$ is difference in surface normal at pixel i; $(\Delta xyz)_i$ is difference in range at i; the weights W_1 and W_2 are normalizing constants used to adjust the relative contributions of surface normal differences and range differences. In this paper, W_1 is taken equal to $1/2$ because the maximum value of $(\Delta surfn)_i$ is 2, while W_1 is taken to be $1/d_{max}$ where d_{max} is the maximum expected range difference.

In cases where a projected model face is slightly shifted from an object in a range image, the difference in range along the shift will be large and may contribute to a large fraction of the similarity measure. To prevent large effects of such registration errors, range differences which exceed d_{max} are set back equal to d_{max}.

The hypothesis generation and verification are performed for all the object models in the model library. The smallest value of S corresponds to the recognized model and the six pose parameters of the object are obtained.

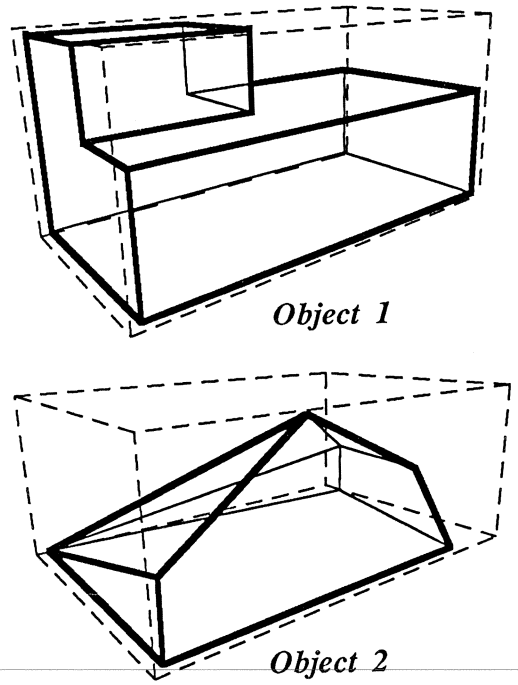

Figure 3: Two object models stored in the model library

Object #1

θ ϕ ψ	T_X, T_Y, T_Z	
	-9.2, 150.1, 95.3	-9.0, 144.7, 89.6
-10.8 -13.0 44.8	191.4	217.7
164.7 -162.8 -134.0	192.1	219.9

Object #2

θ ϕ ψ	T_X, T_Y, T_Z	
	-12.2, 143.6, 92.1	19.7, 149.9, 90.0
15.0 15.0 45.0	103.7	7.8
150.5 -159.8 -15.6	86.5	95.6

Table: Similarity measures for the largest rotation and translation clusters

6 Experimental results and discussion

Two kinds of models shown in Figure 3 were stored in the model library. A synthetic range image was obtained from the second model placed at $\theta = 15°$, $\phi = 15°$, $\psi = 45°$ and $T_X = 20$, $T_Y = 150$, $T_Z = 90$ units.

For the rotation and translation parameter spaces, one thousand data points were obtained. For the cluster detection, the mesh size of the 3-D grids was 60° for the rotation parameter space and 10 units for the translation parameter space. The radii h of the spheres were equal the mesh sizes. The following table summarizes the results obtained for the first and second model, ie. the hypothesized pose parameters found by the LMS cluster detection for the two largest clusters in translation and rotation, leading to four pose hypotheses for each model; the corresponding similarity measures for the four combinations in each model are also shown in the table. The similarity measure S had the smallest value for the second model and for the pose (θ =15.0°, $\phi = 15.0°$, $\psi = 45.0°$),($T_X = 19.7$, $T_Y = 149.9$, $T_Z = 90.0$ units). Note that this choice does not correspond to the largest cluster in translation but to the second largest cluster; the similarity measure allowed us to discard the largest translation cluster, and the estimated pose was very close to the actual pose.

7 Conclusions

We have presented a method for recognizing polyhedral objects in range images based on matching groups of range image features to groups of model features for models described in a library. The more interesting components of the method are the following:

1. The number of considered pairs of image and model features can be reduced by selecting at random only a few of all the possible image features, matching them to appropriate model features.

2. By centering spheres at the nodes of 3-D grids in the parameter spaces, and by estimating new centers using a LMS method, the centers of the spheres are displaced to the centers of the clusters. The largest clusters provide candidate pose estimations of the object in the range image

3. A verification of the candidates based on comparing projected models to the range image proves useful for selecting the clusters in translation and rotation corresponding to the actual object pose. Even

in our simple experiments with synthetic range images, the correct translation cluster was not the largest one.

The experimental results are promising. Obviously, more work is needed. We plan to test the effectiveness of the method with actual range images and noisy synthetic images, and to use larger model libraries; we are studying extensions for recognizing multiple occluding objects.

Acknowledgements

The authors wish to thank Dr. Dong Yoon Kim of the Agency for Defense Development, Taejeon, Korea for discussions on LMS methods.

References

[1] W.E.L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data", Int. J. Robotics Res., vol.3, no.3, pp3-35, Fall 1984.

[2] R.C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system", Int. J. Robotics Res., vol.5, no.3, pp3-26, Fall 1986.

[3] K. Ikeuchi and T. Kanade, "Automatic generation of object recognition programs", Proc. of the IEEE, Vol.76, no.8, pp1016-1035, Aug. 1988.

[4] T. Fan, G. Medioni and R. Nevatia, "Recognizing 3-D objects using surface description", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.11, no.11, pp1140-1157, Nov. 1989.

[5] A.K. Jain and R. Hoffman, "Evidence-based recognition of 3-D objects", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.10, no.6, pp783-802, Nov. 1988.

[6] S. Linnainmaa, D. Harwood, L.S. Davis, "Pose Determination of a Three-Dimensional Object Using Triangle Pairs", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.5, no.5, pp634-647, Sept. 1988.

[7] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recognition, 13 (2), pp111-122, 1981.

[8] D.Y. Kim, J.J. Kim, P. Meer, D. Mintz and A. Rosenfeld, "Robust computer vision: a least median of squares based approach", Proc. Image Understanding Workshop, pp1117-1134, May 1989.

[9] P.J. Rousseeuw, "Least median of squares regression", J. of the American Statistical Association, vol.79, no.388, pp871-880, Dec. 1984.