CONTOUR BASED MATCHING TECHNIQUE FOR 3D OBJECT RECOGNITION

V. Shantaram
AMOST, Inc.
2850 Bowers Avenue
Santa Clara,
CA 95051 U.S.A.
Email: shantaramv@hotmail.com

Madasu Hanmandlu
Faculty of Engineering,
Multimedia University
63100 Cyberjaya
Selangor, Malaysia
Email: madasu.hanmandlu@mmu.edu.my

Abstract

This paper presents a contour matching technique for the identification of an object model corresponding to an observed object from a list of object models from range data. There are three types of edge data associated with the object and the models. These data are utilized in a hierarchical fashion each time employing one type of edge data for pruning the models during the matching. The matching uses quarternion theory and is more suitable for the recognition of symmetric objects. The results are illustrated through simulated examples.

Keywords
Range data, Matching, Quarternions, Euler parameters, Rotationally symmetric surfaces

1. Introduction

Model based methods[1] have been discussed widely in the context of 3D object recognition methods[3]. In the model based recognition problem, there is a given image of an object O which is assumed to be a member of a collection of possible objects $O_1,\ldots,O_n$ known a priori. Our problem is to identify the object O with one of the elements in the list of object models and also to determine its position and orientation. This problem is fundamental to computer vision and has been discussed repeatedly in the literature. Instead of using surface data[2,5-8]. The approach described here differs from Bolles et al[2]. who use cylindrical features, Fisher[6] who uses curvature class and Grimson and Lozano-Perez[7] who use straight line features in the matching process. There are a variety of approaches for dealing with recognition of occluded objects from intensity images. Shwartz and Sharir[10] present an exhaustive list of techniques in this regard. In the present work, we are concerned with the edge data derived from range images after segmentation[8] and our objective is to use this data for short listing a model belonging to the object and then compute its position and orientation. We do not make use of the surface data as reported in the literature2,6 for the purpose of representation of 3D objects.

The proposed approach to the object identification problem is based on the notation of a rigidly embedded curve C defined by the geometry of the object O and fixed at the origin so that when the object undergoes any rotation and translation (i.e., Euclidean motion) the curve C also moves with it. The object models are stored under some generic categories. In order to find the closest match of the given object in the existing models we first shortlist the models by matching the outer boundary of the object in the scene with the outer boundaries (step edges) of stored models and then the inner (semi-step and ramp) edges of the scene object are matched with the inner edges of the short listed models to calculate the necessary rotation and translation information.

Both the outer and inner edges are matched using a fast contour matching technique employing a least squares criterion. Since contour matching involves point sequences, we assume that the scene object and its corresponding model are of the same size.
The organization of this paper is as follows: A quarternion-based contour matching technique is presented in section 2, while the recognition scheme is given in section 3. The results of implementation and conclusions are relegated to sections 4 and 5 respectively.

2. Contour Matching Technique

Here the points on the boundary C' of an observed image is matched with the points on the boundary C of a model object O assuming both C' and C are parameterized by an arc length s. Since C' and C may not start at the same point, they may be separated by an arc length s_0, which we refer to as an offset. The matching calls for the determination of the offset s_0 and Euclidean transformation E for which curves $EC'(s)$ and $C(s + s_0)$ are closest to each other. To be more specific we represent each of the curves C' and C by a sequence of n evenly spaced points. Let these sequences be $(u_j),(v_j)$, $j=1,\ldots,n$ corresponding to the observed contour C' and model contour C respectively. To simplify matching, we initially assume that the whole of the

boundary of O' is visible and no offset calculation is required. Matching then amounts to finding Euclidean motion E which minimizes the least squares distances between the sequences (Eu_j), $j=1,\ldots,n$ and $(v_j), j=1,\ldots,m$; $m \geq n$.
The measure is therefore

$$F = \min_{E} \sum_{j=1}^{n} \left| Eu_j - v_j \right|^2 \qquad (1)$$

For proper matching we also translate O' so that the centroid of the contour lies at the origin, such that

$$\sum_{j=1}^{n} u_j = 0 \qquad (2)$$

Next, we express Eu as $Ru + a$ where R is a rotation matrix and a is a translation vector. In view of this, the measure (1) becomes:

$$F = \min_{R,a} \sum_{j=1}^{n} \left| Ru_j + a - v_j \right|^2 \qquad (3)$$

2.1 Computation of Translation

To obtain the translation parameter $\boldsymbol{a}$ consider the above measure and expand to obtain

$$F = \min_{R,a} \left[\sum_{j=1}^{n} \left| v_j \right|^2 + n|a|^2 - 2\sum_{j=1}^{n} a.v_j + \sum_{j=1}^{n} \left| u_j \right|^2 + 2\sum_{j=1}^{n} a.Ru_j - 2\sum_{j=1}^{n} Ru_j.v_j \right] \qquad (4)$$

But,

$$\sum a.Ru_j = a.R\left(\sum u_j\right) = 0 \qquad (5)$$

Here $\boldsymbol{a}$ and $\boldsymbol{R}$ appear independently in $\boldsymbol{F}$ so that we can minimize their contributions separately. To minimize over $\boldsymbol{a}$, we simply use

$$a = \frac{1}{n} \sum_{j=1}^{n} v_j \qquad (6)$$

as this will eliminate the terms involving $\boldsymbol{a}$, i.e.,

$$n|a|^2 \, and \, -2\sum_{j=1}^{n} a.v_j \, .$$

2.2 Computation of Rotation

As $\boldsymbol{a}$ has been found from the model data, this can be subtracted from v_j to force the centroid of the scene data to be located at the origin. Accordingly, the measure $\boldsymbol{F}$ now becomes:

$$F = \min_{R} \sum_{j=1}^{n} \left| Ru_j - v'_j \right|^2 \qquad (7)$$

where,

$$v_j = v_j - a \qquad (8)$$

Having removed $\boldsymbol{a}$ from minimization process, we minimize the measure with respect to rotation $\boldsymbol{R}$ which relates v_j' to u_j by the following equation

$$v_j = Ru_j \qquad (9)$$

Instead of following the approach[10] for finding $\boldsymbol{R}$, which minimizes $\boldsymbol{F}$, we make use of quarternions for representing a rotation in 3D space. As we know that any three-dimensional vector $\boldsymbol{v}$ is identified with quarternion $\boldsymbol{(o, v)}$ and the product $\boldsymbol{Rv}$ can be identified as the product of two quarternion:

$$Rv = q \otimes v \otimes \bar{q} \qquad (10)$$

This result is used to minimize $\boldsymbol{F}$ as shown below:

2.3 Minimization of $\boldsymbol{F}$ using Quarternions

Consider the measure $\boldsymbol{F}$ and treat u_j and v_j' as quarternions in eqn.(7)

$$F = \min_{R} \sum_{j=1}^{n} \left| Ru_j - v_j \right|^2 \qquad (11)$$

Using the relations (10) and (11), we obtain

$$F = \min_{q} \sum_{j=1}^{n} \left| q * u_j * \bar{q} - v'_j \right|^2 \qquad (12)$$

Since $|q|^2 = 1$, we can multiply both sides with $|q|^2$ resulting in

$$F = \min_{q} \sum_{j=1}^{n} \left| q * u_j - v'_j * q \right|^2 \qquad (13)$$

In view of (A3) in the Appendix, (13) can be written as;

$$F = \min_{q} \sum_{j=1}^{n} \left| u_j^{-} q - v_j'^{+} q \right|^2 \qquad (14)$$

$$= \min_{q} \sum_{j=1}^{n} \left| Aq \right|^2 \qquad (15)$$

where

$$A = u_j^- - v_j^{'+}$$

$$F = \min_q \sum_{j=1}^{n} q^t A^t A q = q^t B q \qquad (16)$$

where

$$B = \sum_{j=1}^{n} A^t A$$

The solution to the above minimization problem is then reduced to finding the eigenvalues of ***B***. The minimum eigenvalue of ***F*** is given by the smallest eigenvalue of ***B***. Thus our error measure due to rotation is

$$F = \mu_{\min}(B) \qquad (17)$$

The eigenvector corresponding to this minimum eigenvalue is the quarternion of interest. From the elements of quarternion, ***R*** can be computed as follows[11]:

$$R = \begin{bmatrix} a_0^2 + a_1^2 - a_2^2 - a_3^2 & 2(a_1a_2 - a_0a_3) & 2(a_1a_2 + a_0a_2) \\ 2(a_1a_2 + a_0a_3) & a_0^2 - a_1^2 + a_2^2 - a_3^2 & 2(a_2a_3 - a_0a_1) \\ 2(a_1a_3 - a_0a_2) & 2(a_2a_3 + a_0a_1) & a_0^2 - a_1^2 - a_2^2 + a_3^2 \end{bmatrix} \qquad (18)$$

where a_0, a_1, a_2, a_3 are the elements of the quarternion.

If ***O'*** is partially occluded, then we have to match the sequence (u_j), $j=1,\ldots,n$ to each of the contiguous sub-sequences (v_{j+d}), $j=1,\ldots,n$ of the circular sequence (v_j), $j=1,\ldots,m$, where ***d*** is the offset in starting points of two sequences.

Here we can appropriately assume ***m > n***, for otherwise the partial periphery of ***O'*** cannot match ***O***. For each ***d***, (11) can be written as

$$F_d = \min_R \sum_{j=1}^{n} \left| R u_j - v'_{j+d} \right|^2 \qquad (19)$$

for $d = 0,1,\ldots,m-1$

and accordingly (14) can be expressed as

$$F_d = \min_q \sum_{j=1}^{n} \left| u_j^- q - v_{j+d}^{'+} q \right|^2 \qquad (20)$$

From (20), we can fix A as

$$A = u_j^- - v_{j+d}^{'+}$$

Thus the elements in $v_j^{'+}$ have to be shifted by ***d***. The rest of the procedure consists of finding ***B***, ***F*** and ***R*** from (16), (17) and (18) respectively.

3. The Proposed Recognition Scheme

In our model based recognition scheme, models of objects are stored in the database through their outer boundaries and the inner edges. Recognition of an object takes place by matching the outer boundary and the inner edges with the corresponding outer boundary and inner edge of one of the models. Matching with both the outer boundary and inner edges is necessary because two different 3D objects may have the same outer boundary but different inner edges.

We have implemented the segmentation procedure[8] in two parts, one generates the outer boundary, which consists of step edges and the other generates the semi-step and roof edges (Figs.1 and 2). In this procedure, first equidistant contours (EDCs) are found by slicing the range data at equal increments of depth. Next, three types of critical points are found. The first type yields the step edges in which the points separate the object from the background. The second type yields the semi-step edges where two surfaces appear to meet but they do not. The third type yields the ramp edges at which two surfaces meet. These are found using by gradient operator[8]. This procedure can be implemented to extract outer boundary and inner edges simultaneously.

To reduce the computational effort, first only the outer boundary of the input segmented scene is matched with the outer boundaries of all the models. Using boundary matching discussed in the previous sections, we find the minimum value of error measure for best possible match between each model and the scene. The minimum error measure is simply the minimum eigenvalue of ***B***. Thus for each model outer boundary we get a value of error measure, and the quaternion which provides rotation is the eigenvector corresponding to the minimum eigenvalue. The models whose outer boundary yields the minimum error is selected, and the inner edges of this model are matched with the inner edges of input segmented image to complete the recognition process which is based again on the same error measure, and to evaluate the position and orientation of the input segmented scene with respect to identified model.

4. Results of Implementation

We have considered range images of three different objects for our study. These range images are passed to the segmentation routine. The segmentation is done in two stages with the first stage giving step edges or the outer boundary and the second stage giving semi-step and roof edges. The range images and the segmented images after two stages of segmentation routine are shown in Figs.3-5.

The scene data is generated by applying rotation transformation matrix to the corresponding model data. We have generated 3 input scenes for our study. Thus, input scene 1 is obtained from model 2 after it is rotated by an angle of 30^0 about the Z-axis. Input scene 2 is obtained from model 1 after it is rotated by an angle of 180^0 about the X-axis. Input scene 3 is obtained from model 3 by rotating it by an angle of 165^0 about the Z-axis. Thus there is no translation between the scene and model data. These input scenes are shown in Figs.3(b),4(b),5(b).

In the case study, objects were subjected to pure rotation, and no translation component was involved. It may be mentioned that in the proposed technique, the translation component may be computed first before proceeding with the computation of rotation component. In this sense the two computations are delinked.

The centroid of the first image is $(84,24,176)$.

This centroid is then shifted to the origin. The resulting data forms the model. Next, a rotational matrix is used to create the scene 1 data. The actual rotation matrix used to do the same is given by

$$\begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Results of matching the outer boundary of input scene 1 with various models are given in Table 1.

So, the short listed model is 2. The computed rotation matrix is

$$R = \begin{bmatrix} 0.8072 & -0.4673 & -0.0008 \\ 0.46773 & 0.8072 & -0.0003 \\ 0.0008 & -0.0002 & 0.9327 \end{bmatrix}$$

Table 1: Results of Contour matching

Scene No.	Model No.	Quartenions a_0, a_1, a_2, a_3	Error measure **F**
Scene 1	Model 1	0.6451535 0.1293007 0.4498127 0.0994000	$7.2142\text{x}10^5$
Scene 1	Model 2	0.9329098 -0.0000367 0.0002674 0.250178	$8.345\text{x}10^1$
Scene 1	Model 3	0.7753804 0.3937158 0.1153466 0.0764765	$5.50271\text{x}10^5$

Similar studies are made for the other two images. Now, consider the second image whose centroid is $(61,-64,-191)$. The actual rotation matrix used to create the scene 2 data is:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Results of matching the outer boundary of input scene 2 with the various models are given in Table 2.

Table 2: Results of Contour matching

Scene No.	Model No.	Quartenions a_0, a_1, a_2, a_3	Error measure **F**
Scene 2	Model 1	0.0000 1.0000 0.0000 0.0000	$0.0000\text{x}10^1$
Scene 2	Model 2	0.0077303 0.6522407 0.0718296 0.4707478	$7.2277\text{x}10^5$
Scene 2	Model 3	0.2079023 -0.5784629 0.3961512 -0.2090081	$7.49116\text{x}10^5$

So the short listed model is 1. The computed rotation matrix is

$$R = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 0.0 \\ 0.0 & 0.0 & -1.0 \end{bmatrix}$$

Finally, we consider the third image whose centroid is $(-74,-45.199)$. The actual rotation matrix used to create the scene 3 data is:

$$R = \begin{bmatrix} -0.966 & 0.259 & 0 \\ 0.259 & -0.966 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Results of matching the outer boundary of input scene 3 with the various models are given in Table 3.

Table 3: Results of Contour matching

Scene No.	Model No.	Quartenions a_0, a_1, a_2, a_3	Error measure **F**
Scene 2	Model 1	0.0000 1.0000 0.0000 0.0000	$6.7309\text{x}10^5$
Scene 2	Model 2	0.0077303 0.6522407 0.0718296 0.4707478	$5.5084\text{x}10^5$
Scene 2	Model 3	0.2079023 -0.5784629 0.3961512 -0.2090081	$5.7299\text{x}10^1$

So the short listed model is 3. The computed rotation matrix is

$$R = \begin{bmatrix} -0.9497 & 0.2538 & -0.0006 \\ -0.2538 & -0.9497 & 0.0004 \\ -0.0005 & 0.0005 & 0.9830 \end{bmatrix}$$

5. Conclusions

In this paper, the boundary matching technique of Schwartz and Sharir[10] has been reformulated to encompass the recognition of 3D objects from the range images. However, the basic approach is quite different. In our scheme, outer boundary of the scene data is matched with the outer boundaries of all the listed models. The computed values of rotation are quite comparable with the actual rotation values. The errors may be attributed to inaccuracies involved in the computation of eigenvalues and eigenvectors.

We have used edges in our model-based recognition scheme. The edge data reduces the total amount of range data needed for the purpose of object recognition. A recognition system based on edge description, however, is not robust and cannot distinguish between very similar objects. Thus to make a robust system which is also fast, both edge and surface information should be used. The edge information can be effectively used for short-listing of models for the purpose of matching, and surface information can be used for final recognition and the computation of rotation and translation components of scene data with respect to the identified model. The proposed approach can only be used for identifying extremal boundaries of curved surfaces, as the edge data of arbitrary object may be different in different views. The technique would work only with the points that do not move as the viewpoint changes.

References

[1] P.J. Besl and J.C. Jain, "Three Dimensional Object Recognition", Computing Surveys Vol.17, 1985, pp.75-145.

[2] R.C. Bolles, P.D. Horaud, and MJ. Hannah, "3DPO : A three dimensional part orientation system", Proceedings of 8th IJCAII, 1983, pp.116-120.

[3] R.T. Chin, and C.R. Dyer, "Model-based recognition in robot vision",ACM Computing Surveys, Vol.18, 1986, pp.67-108.

[4] J.C.K. Chou and M.Kamel, "Quaternions approach to solve the kinematic equation of rotation, Aa Ax = Ax Ab of a robotic manipulator", Proceedings of IEEE Conference on Robotics and Automation, 1988, pp.656-662.

[5] O.D. Faugeras and M.Hebert, "The Representation, recognition and locating of 3-D objects",The International Journal of Robotics Research, Vol.3, 1986, pp.27-52.

[6] R.B. Fisher, "From Surface to Objects: Computer Vision and three-dimensional analysis", 1989, John Willey & Sons Ltd.

[7] W.E.L. Grimson and T. Lozano-Perez, "Model based recognition and localization from sparse range data or tactile data", The International Journal of Robotics Research, Vol.3, 1984, pp.3-35.

[8] M. Hanmandlu, C. Rangaiah and K.K. Biswas, "Quadrics-based matching technique for 3D object recognition",Image and Vision Computing, Vol.10, 1992, pp.578 –588.

[9] R.P. Paul, "Robot Manipulators: Mathematics, Programming and Control", MIT Press, Cambridge, MA, 1981.

[10] J. Schwartz and M. Sharir, " Identification of partially occluded objects in two and three dimensions by matching noisy characteristic curves, The International Journal of Robotics Research, Vol.6, 1986, pp.29-43.

[11] Z. Zhang and O.D. Faugeras, "Determining motion from 3D line segments: A comparative study", Image and Vision Computing, Vol.9, 1991, pp.10-19.

Fig.1 Segmented range image of the object with labels (a) step edge; (b) roof edge; (c) semi-step edge

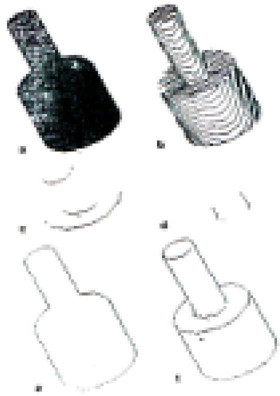


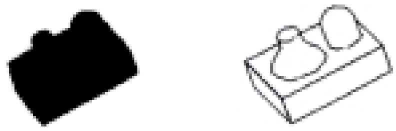
Fig.2 Segmentation at various stages. (a) raw image; (b) equi depth contours; (c) roof edges; (d) semi-step edges; (e) step edges; (f) segmented range image

(a) (b)

Fig.3 (a) range image (b) segmented image

(a) (b)

Fig.4 (a) range image (b) segmented image



(a) (b)

Fig.5 (a) range image (b) segmented image

Appendix

A quarternion α is defined as a complex vector[4]

$$\alpha = \alpha_0 + a_1 i + a_2 j + a_3 k \qquad \text{(A1)}$$

such that $\alpha = a_0 + a$

If α and β are two quarternions, their multiplication (x) is defined as:

$$\alpha(x)\beta = (a_0 + a)(x)(b_0 + b)$$
$$= a_0(x)b_0 + a_0(x)b + b_0(x)a + a(x)b \qquad \text{(A2)}$$

Separating scalar and vector products we can rewrite it as:

$$\begin{bmatrix} c_0 \\ c \end{bmatrix} = \begin{bmatrix} a_0 & -a^t \\ a & a_0 I + \tilde{a} \end{bmatrix} \begin{bmatrix} b_0 \\ b \end{bmatrix}$$
$$= \begin{bmatrix} b_0 & -b^t \\ b & b_0 I - \tilde{b} \end{bmatrix} \begin{bmatrix} a_0 \\ a \end{bmatrix}$$
$$= \alpha^+ \beta = \beta^- \alpha \qquad \text{(A3)}$$

where, the signs '+' and '-' on α and β correspond to the signs '+' and '-' attached to $\tilde{a}$ and $\tilde{b}$ respectively, and $\tilde{a}$ defines the skew symmetric matrix given by

$$\tilde{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \qquad \text{(A4)}$$