Free-Form 3-D Object Recognition at Multiple Scales

Farzin Mokhtarian, Nasser Khalili and Peter Yuen
Centre for Vision, Speech, and Signal Processing
School of Electronic Engineering, IT and Mathematics
University of Surrey, Guildford, GU2 7XH, UK
F.Mokhtarian@ee.surrey.ac.uk
http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/

Abstract

The recognition of free-form 3-D objects using multi-scale features recovered from 3-D models, and based on the geometric hashing algorithm and global verification is presented. The feature points on the object are detected by smoothing its surface after construction of semigeodesic coordinates at each point (mesh vertex). This technique is the generalisation of the CSS method which is a powerful shape descriptor expected to be in the MPEG-7 standard. Smoothing is used to remove noise and to select multi-scale feature points to add to the efficiency and robustness of the system. The local maxima of Gaussian and mean curvatures are selected as feature points. Furthermore the torsion maxima of the zero-crossing contours of Gaussian and mean curvatures are also selected as feature points. Recognition results are demonstrated for rotated and scaled as well as partially occluded objects. In order to confirm the match, 3D translation, rotation and scaling parameters are used for verification and results indicate that our technique is invariant to those transformations.

1 Introduction

Object recognition is a major task in computer vision. Surface curvature provides a unique view-point invariant description of local surface shape. Differential geometry [4] provides several measures of curvature, which include Gaussian and mean curvatures. In this paper the recognition of free-form 3-D objects using multi-scale features recovered from 3-D models, and based on the geometric hashing technique and global verification is addressed. The model information is indexed into a hashtable using minimal transformation invariant features. The feature points on the object are detected by convolving local parametrisations of the surface with 2-D Gaussian filters [10, 18]. The recognition of a partially occluded object in a scene amounts to the discovery of a match between a subset of the scene interest features and a subset of the interest features of some model object.

The organisation of this paper is as follows. Section 2 gives a brief overview of previous work on 3-D object representation, recognition and matching. Section 3 reviews the relevant theory from differential geometry and explains how a multi-scale shape description can be computed for a free-form 3-D surface. Section 4 explains curvature estimation and detection of local maxima. Section 5 describes our system for robust 3-D model-based object recognition. Section 6 presents results and discussion. Section 7 contains the concluding remarks.

2 Literature Survey

This section presents a survey of previous work in representation and recognition of 3-D surfaces. Sinha and Jain [13] provide an overview of geometry based representations derived from range data of objects [2]. Volumetric representations rely on representing objects in terms of generalised cylinders, superquadrics, set-theoric combinations of volume primitives as in constructive solid geometry (CSG) or spatial occupancy [14, 1, 11]. However, it may not be possible to express objects with free-form surfaces using for example, superquadric primitives. Although there are several methods available to model a surface, triangular meshes are the simplest and most effective form of polygons to cover a free-form surface. The common types of polygonal meshes include the triangular mesh [5] and the four sided spline patches. Triangular meshes have been utilised in our work.

In *volumetric diffusion* [6] or level set methods [12], an object is treated as a filled area or volume. The object is then blurred by subjecting it to the diffusion equation. The boundary of each blurred object can then be defined by applying the Laplacian operator to the smoothed area or volume. The major shortcoming of these approaches is lack of local support. In other words, the entire object data must be available. This problem makes them unsuitable for object recognition in presence of occlusion. A form of 3-D surface smoothing has been carried out in [16] but this method has drawbacks since it is based on weighted averaging using neighbouring vertices and is therefore dependent on the underlying triangulation. The smoothing of 3-D surfaces is a result of the diffusion process [17].

A number of matching topics have been recognised by researchers as important in 3-D object recognition [3]. These are related to object shape complexity, rigid and flexible objects and occlusion. The success of existing object recognition systems is because of the restrictions they impose on the classes of geometric objects. However, few systems can handle arbitrary surfaces with very few restrictive assumptions about their geometric shapes.

Object recognition is achieved by matching features derived from the scene with stored object model representations. Efficient algorithms were developed for the recognition of flat rigid objects based on the Geometric Hashing technique in [7]. The technique was also extended to the recognition of arbitrary rigid 3-D objects from single 2-D images [8]. Stein and Medioni [15] and Flynn and Jain [3] have also employed geometric hashing for 3-D object recognition. In a Geometric Hashing technique the model information is indexed into a hash table using minimal transformation feature points. This technique determines for a given scene's minimal feature set a corresponding feature set on one of the models, by considering only the other scene features which *vote* for the correct interpretation. The advantages of the Geometric Hashing technique over the others is the independent processing of the model and scene information as well as its ability to process all the model objects simultaneously, and its quick recognition time.

3 Semigeodesic Parametrisation

Free-form 3-D surfaces are complex hence, no global coordinate system exists on these surfaces which could yield a natural parametrisation of that surface. Studies of local properties of 3-D surfaces are carried out in differential geometry using local coordinate systems called *curvilinear coordinates* or *Gaussian coordinates* [4]. Each system of curvilinear coordinates is introduced on a patch of a regular surface referred to as a *simple sheet*. A simple sheet of a surface is obtained from a rectangle by stretching, squeezing, and bending but without tearing or gluing together. Given a parametric representation $\mathbf{r} = \mathbf{r}(u,v)$ on a local patch, the values of the parameters u and v determine the position of each point on that patch. Construction and implementation of semigeodesic coordinates in our technique is described in [10, 19].

4 Curvature Estimation

This section presents techniques for accurate estimation of Gaussian and mean curvatures at multiple scales on smoothed free-form 3-D surfaces. Differential geometry provides several measures of curvature, which include Gaussian and mean curvatures [4]. Consider a local parametric representation of a 3-D surface

$$\mathbf{r} = \mathbf{r}(u,v) = (x(u,v),\ y(u,v),\ z(u,v))$$

Gaussian curvature K exists at regular points of a surface of class C_2. When $\mathbf{r}(u,v)$ corresponds to semigeodesic coordinates, K is given by:

$$K = \frac{b_{uu}b_{vv} - {b_{uv}}^2}{x_v^2 + y_v^2 + z_v^2} \tag{1}$$

where subscripts denote partial derivatives, and

$$b_{ij} = \frac{Ax_{ij} + By_{ij} + Cz_{ij}}{\sqrt{A^2 + B^2 + C^2}}$$

where each of i and j can be either u or v, and

$$A = y_u z_v - z_u y_v \qquad B = x_v z_u - z_v x_u \qquad C = x_u y_v - y_u x_v.$$

Mean curvature H also exists at regular points of a surface of class C_2. Again, when $\mathbf{r}(u,v)$ corresponds to semigeodesic coordinates, H is given by:

$$H = \frac{b_{vv} + (x_v^2 + y_v^2 + z_v^2)b_{uu}}{2(x_v^2 + y_v^2 + z_v^2)} \tag{2}$$

Gaussian and mean curvatures are invariant to arbitrary transformation of the (u,v) parameters as well as rotations and translations of the surface. On smoothed surfaces of 3-D objects, the procedure for estimating the Gaussian and mean curvatures are as follows. For each point $p(x(u,v),\ y(u,v),\ z(u,v))$ of the surface, the corresponding local neighbourhood data is convolved with the partial derivatives of the Gaussian function $G(u,v,\sigma)$. Finally, curvature values on a 3-D surface are estimated by substituting these values into equations (1) and (2), respectively.

4.1 Local Curvature Maxima

Local maxima of Gaussian and mean curvatures are significant and robust feature points on smoothed surfaces since noise has been eliminated from those surfaces. The process of recovery of the local maxima is identical for Gaussian and mean curvatures. Every vertex V of the smoothed surface is examined in turn. The neighbours of V are defined as vertices which are connected to V by an edge. If the curvature value of V is higher than the curvature values of all its neighbours, V is marked as a local maximum of curvature. Curvature maxima can be utilised by later processes for robust surface matching and object recognition with occlusion.

4.2 Torsion Maxima of Curvature Zero-Crossing Contours

This section briefly reviews the computation of torsion. Torsion is the instantaneous rate of change of the osculating plane with respect to the arc length parameter. The osculating plane at a point P is defined to be the plane with the highest order of contact with the curve at P [4]. The set of points of a space curve are the values of a continuous, vector-valued, locally one-to-one function

$$r(u) = (x(u),\ y(u),\ z(u))$$

where $x(u)$, $y(u)$ and $z(u)$ are the components of $r(u)$, and u is a function of arc length of the curve. In order to compute torsion τ at each point of the curve, it is then expressed in terms of the derivatives of $x(u)$, $y(u)$ and $z(u)$. In case of an arbitrary parametrisation, torsion is given by,

$$\tau = \frac{\dot{x}(\ddot{y}\check{z} - \ddot{z}\check{y}) - \dot{y}(\ddot{x}\check{z} - \ddot{z}\check{x}) + \dot{z}(\ddot{x}\check{y} - \ddot{y}\check{x})}{(\dot{y}\ddot{z} - \dot{z}\ddot{y})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{x})^2 + (\dot{x}\ddot{y} - \dot{y}\ddot{x})^2} \tag{3}$$

where $\dot{x}(u)$, $\ddot{x}(u)$ and $\check{x}(u)$ are the first, second and third derivatives of $x(u)$ respectively.

5 Robust Free-Form 3-D Object Recognition

The geometric hashing technique for model based object recognition was introduced by Lamdan and Wolfson [7, 8]. Stein and Medioni [15] as well as Flynn and Jain [3] have also employed geometric hashing for 3-D object recognition. Our 3-D object recognition system employs a version of the geometric hashing method followed by global verification to robustly recognise 3-D objects.

5.1 Geometric Hashing

The matching stage of the algorithm uses the hash table prepared in the off-line stage. Given a scene of feature points, one tries to match the measurements taken at scene points to those memorised in the hash table. On smoothed surfaces of 3-D objects, the procedure for indexing data into the hash table is as follows. For each 3-D object in the database:

1. The local maxima of Gaussian curvature are selected as feature points. Furthermore, local maxima of mean curvature and the torsion maxima of the zero-crossing contours of Gaussian and mean curvatures are also selected as feature points.
2. Choose an arbitrary ordered triplet of non-collinear points A, B and C to form a triangle ABC. Denote the curvature values of points A, B and C by k_a, k_b and k_c, and the edge lengths AB, BC and AC as d_1, d_2 and d_3, respectively, see Figure 1.

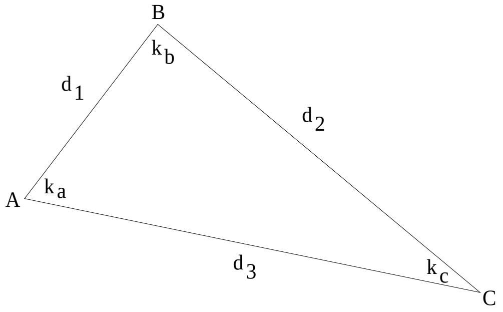


Figure 1: Triplet of non-collinear points A, B and C

Select the maximum curvature value and edge length. Let k_a and d_1 be maximum curvature value and edge length, then calculate the indexed value IV for the hash table as,

$$IV = \frac{k_a}{k_b}\frac{k_a}{k_c}\frac{d_2 + d_3}{d_1} \tag{4}$$

3. Go back to step (2) and repeat the procedure for different triplets of feature points. Note that some of these new selected points may have already been chosen in previous stages.

   We now have produced a hash table with all the data indexed into its memory from a given database. Given a scene of feature points from a 3-D object, we try to match the index value IV as well as the individual ratios to those memorised in the hash table. Notice that the input 3-D object can either be complete or incomplete. Thus, given a 3-D object in a scene, the matching procedure is as follows:
4. Repeat steps (1) to (3) above, and then for each indexed value IV check the appropriate entry in the hash-table. Tally a vote for each model which appears at that location.

5. If several objects score large number of votes close to each other, then the most likely candidate will be chosen using global verification applied at next stage.

The voting is done simultaneously for all models in the hash table. The overall recognition time is dependent on the number of feature points in the scene. Our aim is to significantly reduce the number of feature points that are due to noise. This is achieved through the use of multi-scale feature points or feature points which survive a few iterations of smoothing.

5.2 Global Verification

In general the voting scheme may yield more than one candidate solution with very close scores from the geometric hashing stage. In this case we use a threshold to select the most likely models.

From the 3 points of the scene model selected for matching, as discussed in the previous section, another point is also determined which is the centre of gravity of all 3 points in the space. Let $P_1(x_{1p}, y_{1p}, z_{1p})$, $P_2(x_{2p}, y_{2p}, z_{2p})$ and $P_3(x_{3p}, y_{3p}, z_{3p})$ be the 3 non-collinear points selected from the model object and $P_4(x_{4p}, y_{4p}, z_{4p})$ be a point in the 3-D space which is the centre of gravity of P_1, P_2 and P_3. We then form a plane P in the space from all of these points. The same procedure is also applied to the object in the scene. Let $Q_1(x_{1q}, y_{1q}, z_{1q})$, $Q_2(x_{2q}, y_{2q}, z_{2q})$, $Q_3(x_{3q}, y_{3q}, z_{3q})$ and $Q_4(x_{4q}, y_{4q}, z_{4q})$ be the points in the 3-D space. Point Q_4 is the centre of gravity of points Q_1, Q_2 and Q_3, thus a plane Q is also formed. We linearise the problem to simplify solution, hence, the linear equations for the transformation, mapping model points to scene points are given by [9],

$$\begin{bmatrix} x_{1p} & y_{1p} & z_{1p} & 1 \\ x_{2p} & y_{2p} & z_{2p} & 1 \\ x_{3p} & y_{3p} & z_{3p} & 1 \\ x_{4p} & y_{4p} & z_{4p} & 1 \end{bmatrix} \cdot \begin{bmatrix} a & e & m \\ b & f & n \\ c & g & p \\ d & h & q \end{bmatrix} = \begin{bmatrix} x_{1q} & y_{1q} & z_{1q} \\ x_{2q} & y_{2q} & z_{2q} \\ x_{3q} & y_{3q} & z_{3q} \\ x_{4q} & y_{4q} & z_{4q} \end{bmatrix} \tag{5}$$

Note that this approach is employed in order to obtain a quick and approximate solution which is sufficient for verification. From the set of linear equations (5), one can solve for the twelve parameters a, b, c, d, e, f, g, h, m, n, p and q. In order to verify match, 3D translation, rotation and scaling will be used to determine global consistency. The translation parameters are (d, h, q). Let γ, β and α be the angles in the x, y and z directions for the rotation of the plane P in 3D space. The 3D rotation matrices about x-axis, y-axis and z-axis denoted $R_x(\gamma)$, $R_y(\beta)$ and $R_z(\alpha)$, respectively, are given by

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \tag{6}$$

$$R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \tag{7}$$

$$R_z(\alpha) \ = \begin{bmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

The columns (and the rows) of matrices $R_x(\gamma)$, $R_y(\beta)$ and $R_z(\alpha)$ are mutually perpendicular unit vectors and they have determinant of 1, so they are orthogonal. Therefore the rotation parameters (γ, β, α) can be obtained from products of $R_z(\alpha).R_y(\beta).R_x(\gamma)$ and also the solution of equation (5),

$$\begin{bmatrix} a \\ e \\ m \\ n \\ p \end{bmatrix} = \begin{bmatrix} cos\alpha \ cos\beta \\ sin\alpha \ cos\beta \\ -sin\beta \\ cos\beta \ sin\gamma \\ cos\beta \ cos\gamma \end{bmatrix} \tag{9}$$

For the scaling factor $\Re$, the distances from the centre of gravity points P_4 and Q_4 to their corresponding 3 points are measured and the shortest distances for each object are selected. Let r_1 and r_2 be the shortest distances selected from the model object and the scene object, then their ratio is the scaling factor $\Re$.

$$\Re = \frac{r_1}{r_2}$$

A number of model objects with close high scores are selected for the global verification stage. The hash table yields many candidate matches for each selected model. For each of these candidates, seven global transform parameters are estimated, using the procedure described earlier. The candidates are compared and if their corresponding parameters are compatible, they are clustered together. The largest cluster then indicates the largest group of globally consistent matches for each model. The model object with the largest cluster is then chosen as the most likely object present in the scene.

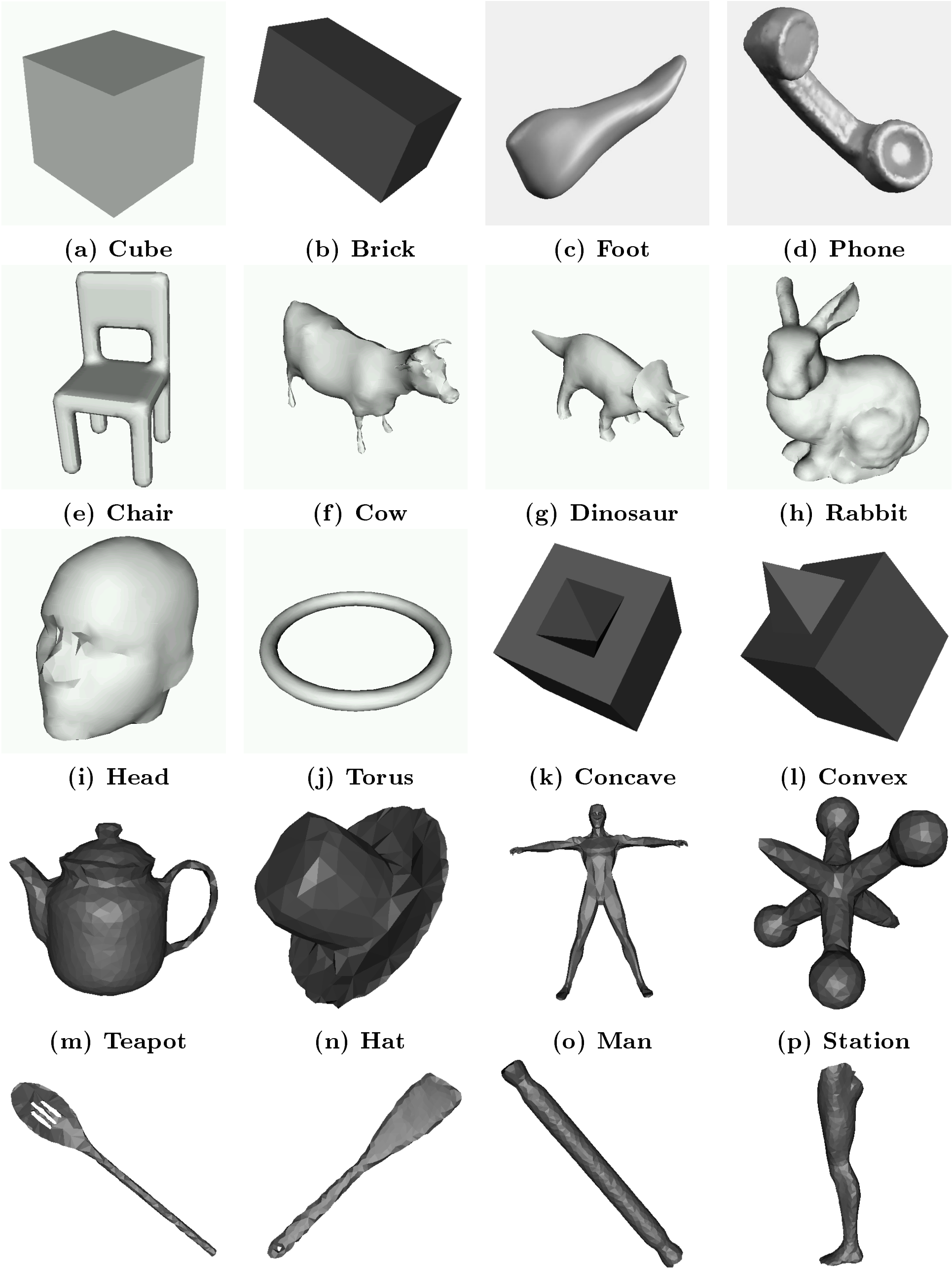
6 Results and Discussion

The smoothing routines were implemented entirely in C++. Each iteration of smoothing of a surface with 1000 vertices takes about 0.5 second of CPU time on an UltraSparc 170E. The diffusion and curvature estimation results for 3-D surfaces were given in [10, 18]. Animation of surface diffusion as well as feature recovery can be observed at our web site:

`http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/`

This section presents the matching results of the system applied to free-form 3-D surfaces in an object database. There are 20 different objects in our database. All of those are shown in Figure 2. In most cases, the data corresponds to real objects.

The first experiment consisted of applying arbitrary amounts of scaling and 3-D rotation to the database objects, and to determine whether they can be recognised correctly by the system. All objects were recognised correctly by the system. In the first stage, geometric hashing was applied to the input object. If one of the



(a) **Cube** (b) **Brick** (c) **Foot** (d) **Phone**

(e) **Chair** (f) **Cow** (g) **Dinosaur** (h) **Rabbit**

(i) **Head** (j) **Torus** (k) **Concave** (l) **Convex**

(m) **Teapot** (n) **Hat** (o) **Man** (p) **Station**

(q) **Spoon** (r) **Spatula** (s) **Roller** (t) **Leg**

Figure 2: Free-form 3-D objects used for matching experiments

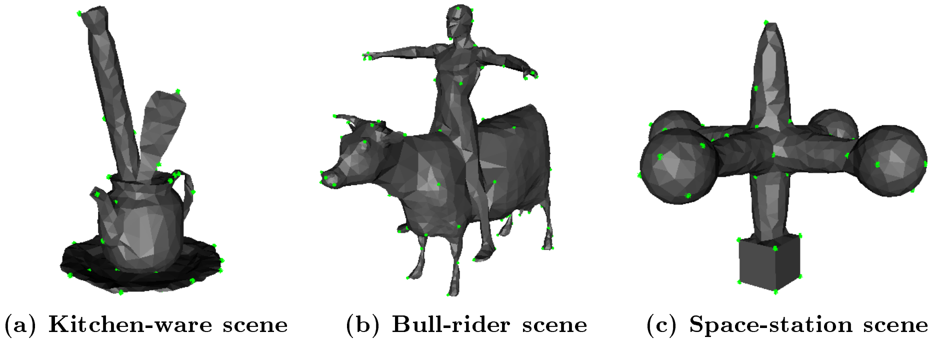

(a) **Kitchen-ware scene** (b) **Bull-rider scene** (c) **Space-station scene**

Figure 3: Three scenes with extracted features

models $\mathcal{M}$ received a vote count that was substantially higher than the vote counts for all other objects, then $\mathcal{M}$ was selected as the correct object and the system terminated. Otherwise, two or more models received high vote counts that were relatively similar. In this case, the system applied global verification only to the surviving models in order to select one of them.

The second experiment made use of incomplete surfaces which were again subjected to arbitrary amounts of scaling and 3-D rotation. In order to obtain incomplete surfaces, up to 60% of connected vertices were removed from database objects. As in the previous experiment, for each input object, the system applied geometric hashing to all database models followed by global verification to the surviving models. Again, all input objects were correctly recognised by the system.

In the third experiment, three complex scenes were created each consisting of two or more objects. Figure 3 shows the three scenes with their extracted features. These include the *kitchen-ware* scene, the *bull-rider* scene, and the *space-station* scene. The *kitchen-ware* scene contains a dish, a kettle, a spatula and a roller. The *bull-rider* scene contains a cow and a rider. The *space-station* scene contains a space-station and a space-ship attached to it.

As in the earlier experiments, the system applied geometric hashing to all database models followed by global verification to the surviving models. In the *kitchen-ware* scene, *dish* scored highest with other scene objects also receiving high scores. In the *bull-rider* scene, *cow*, *dinosaur* and *rider* received the three highest scores respectively. In the *space-station* scene, the station itself received the highest score.

7 Conclusions

The recognition of free-form 3-D objects using 3-D models under different viewing conditions based on the geometric hashing algorithm and global verification was presented. The matching stage of the algorithm used the hash table prepared in the off line stage. The feature points on the object were detected by convolving local parametrisations of the surface with 2-D Gaussian filters iteratively. Smoothing was used to reduce number of feature points. The surface Gaussian and mean curvature values were estimated accurately at multiple scales. In our technique the local maxima of Gaussian and mean curvatures were selected as feature points. Furthermore, the torsion maxima of the zero-crossing contours of Gaussian and mean curvatures were also selected as feature points. This technique was also shown to be useful for partially occluded objects. In order to verify match, 3D translation, rotation and scaling parameters were used for global verification and results indicated that our technique is invariant to those transformations.

References

[1] T W Chen and W C Lin. A neural network approach to CSG-based 3-D object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(7):719–726, 1994.

[2] C Dorai and A K Jain. Shape spectrum based view grouping and matching of 3-d free-form objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(10):1139–1146, 1997.

[3] P J Flynn and A K Jain. Three dimensional object recognition. *in Handbook of Pattern Recognition and Image Processing (T.Y.Young, ed.)*, 2:497–541, 1994.

[4] A Goetz. *Introduction to differential geometry.* Addison-Wesley, Reading, MA, 1970.

[5] A Hilton, A J Stoddart, J Illingworth, and T Windeatt. Reliable surface reconstruction from multiple range images. In *Proc European Conference on Computer Vision*, pages 117–126, Cambridge, UK, 1996.

[6] J J Koenderink. *Solid shape.* MIT Press, Cambridge, MA, 1990.

[7] Y Lamdan, J T Schwartz, and H J Wolfson. Object recognition by affine invariant matching. In *Proc of CVPR*, pages 335–344, Ann Arbor, Michigan, 1988.

[8] Y Lamdan and H J Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc International Conference on Computer Vision*, pages 238–249, Tampa, Florida, 1988.

[9] F Mokhtarian. A theory of multi-scale, torsion-based shape representation for space curves. *Computer Vision and Image Understanding*, 68(1):1–17, 1997.

[10] F Mokhtarian, N Khalili, and P Yuen. Multi-scale 3-D free-form surface smoothing. In *Proc British Machine Vision Conference*, pages 730–739, 1998.

[11] H Samet. The design and analysis of spatial data structures. *Addison-Wesley*, 1990.

[12] J A Sethian. *Level set methods.* Cambridge University Press, 1996.

[13] S S Sinha and R Jain. Range image analysis. In *Handbook of Pattern Recognition and Image Processing: Computer Vision (T Y Young, ed.)*, volume 2, pages 185–237, 1994.

[14] F Solina and R Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 12:131–147, 1990.

[15] F Stein and G Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Trans Pattern Analysis and Machine Intelligence*, 14:125–145, 1992.

[16] G Taubin. Curve and surface smoothing without shrinkage. In *Proc International Conference on Computer Vision*, pages 852–857, 1995.

[17] B M ter Haar Romeny. *Geometry driven diffusion in computer vision.* Kluwer Academic, 1994.

[18] P Yuen, N Khalili, and F Mokhtarian. Curvature estimation on smoothed 3-d meshes. In *Proc British Machine Vision Conference*, pages 133–142, 1999.

[19] P Yuen, F Mokhtarian, and N Khalili. Multi-scale 3-d surface description: open and closed surfaces. In *Scandinavian Conference on Image Analysis*, pages 303–310, Greenland, 1999.