

Multi-scale free-form 3D object recognition using 3D models

F. Mokhtarian*, N. Khalili, P. Yuen

Centre for Vision, Speech, and Signal Processing, School of Electronic Engineering, Information Technology and Mathematics, University of Surrey, Guildford, Surrey GU2 5XH, UK¹

Received 18 November 1999; revised 1 August 2000; accepted 4 September 2000

Abstract

The recognition of free-form 3D objects using 3D models under different viewing conditions based on the geometric hashing algorithm and global verification is presented. 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. The technique used for feature recovery is the generalisation of the CSS method (IEEE Trans. Pattern Anal. Mach. Intell., 14 (1992) 789–805), which is a powerful shape descriptor expected to be an MPEG-7 standard. Smoothing is used to remove noise and reduce the number of 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 verify match, 3D translation, rotation and scaling parameters are used for verification and results indicate that our technique is invariant to those transformations. Our technique for smoothing and feature extraction is more suitable than level set methods or volumetric diffusion for object recognition applications since it is applicable to incomplete surface data that arise during occlusion. It is also more efficient and allows for accurate estimation of curvature values. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: 3D object recognition; Free-form surface matching; Gaussian and mean curvature maxima

1. Introduction

Object recognition is a major task in computer vision. Surface curvature provides a unique viewpoint invariant description of local surface shape. Differential geometry [8] provides several measures of curvature, which include Gaussian and mean curvatures. Combination of these curvature values enables the local surface type to be categorised.

In this paper the recognition of free-form 3D objects using 3D models based on the geometric hashing technique and global verification is addressed. This technique is useful for partially occluded objects. The model information is indexed into a hash-table using minimal transformation invariant features. The feature points on the object are detected by convolving local parametrisations of the surface with 2D Gaussian filters [23,44]. Some interest features are extracted from the objects, so that both the model object and the observed scene can be represented by sets of these inter-

est features. 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 recognition time depends directly on the complexity of the scene to be recognised.

The paper is concluded with examples showing 3D objects with their local curvature maxima as well as the maxima of torsion of zero-crossing contours of Gaussian and mean curvatures, located and displayed on the surface. Furthermore, the matching results for 3D surfaces with arbitrary shapes in a database will be presented.

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

* Corresponding author. Tel.: +44-1483-876035; fax: +44-1483-876031.

E-mail address: f.mokhtarian@ee.surrey.ac.uk (F. Mokhtarian).

¹ <http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/index.html>.

2. Literature survey

This section presents a survey of previous work in representation and recognition of 3D surfaces. Sinha and Jain [31] provide an overview of geometry-based representations derived from range data of objects. Comprehensive surveys of 3D object recognition systems are presented by Besl and Jain [1], Chin and Dyer [4] and Suetens et al. [37]. Some representation schemes for 3D objects have adopted some form of surface or volumetric parametric models to characterise the shape of the objects. Current volumetric representations rely on representing objects in terms of general cylinders, superquadrics, set-theoretic combinations of volume primitives as in constructive solid geometry (CSG) or spatial occupancy [3,26,28,32]. 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 mesh include the triangular mesh [9] and the four-sided spline patches. Triangular meshes have been utilised in our work.

Polyhedral approximations [6] fit a polyhedral object with vertices and relatively large flat faces to a 3D object. Their disadvantage is that the choice of vertices can be quite arbitrary, which renders the representation not robust. Smooth 3D splines [36] can also be fitted to 3D objects. Their shortcomings are that the choice of knot points is again arbitrary and that the spline parameters are not invariant. *Generalised cones* or *cylinders* [33] as well as *geons* [27] approximate a 3D object using globally parametrised mathematical models, but they are not applicable to detailed free-form objects. *Multi-view* representations [29] are based on a large number of views of a 3D object obtained from different viewpoints, but difficulties can arise when a non-standard view is encountered. In *volumetric diffusion* [16] or level set methods [30], 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 3D surface smoothing has been carried out in Refs. [38,39] 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 3D surfaces is a result of the diffusion process [40]. For parametrisation of a 3D surface other methods have also been studied, such as the asymptotic coordinates [17], isothermic coordinates [8] and global coordinates [2] used for closed, simply connected objects.

A number of matching topics have been recognised by researchers as important in 3D object recognition [7,42]. 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 [18,19]. The technique was also extended to the recognition of arbitrary rigid 3D objects from single 2D images [20]. Stein and Medioni [34] and Flynn and Jain [7] have also employed geometric hashing for 3D 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 that *vote* for the correct interpretation. Other efficient model-based object recognition techniques are the Hough (pose) clustering [21,35], the alignment technique [12] and relational structures [43]. 3D objects have also been modelled as superquadrics with local and global deformations for recognition purposes [32].

Recent patch-based techniques including point signatures [5] and spin images [13] perform well on scenes containing clutter and occlusion. However, these systems have been designed for single range images, and do not generalise to more general 3D surfaces which can be obtained by merging two or more range images. In other words, their effectiveness is limited by the use of information in only one range image.

3. Semigeodesic parametrisation.

Free-form 3D 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 3D surfaces are carried out in differential geometry using local coordinate systems called *curvilinear coordinates* or *Gaussian coordinates* [8]. 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 is described in Refs. [14,23,45,46].

4. Curvature estimation

This section presents techniques for accurate estimation of Gaussian and mean curvatures at multiple scales on smoothed free-form 3D surfaces. Differential geometry provides several measures of curvature, which include Gaussian and mean curvatures [8]. Note that Gaussian and mean curvatures are closely related to principal curvatures. As a result, principal curvatures can be used as an alternative. We chose to use Gaussian and mean curvatures since they are non-directional quantities and more intuitive. Consider a local parametric representation of a 3D surface

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

with coordinates u and v , where

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

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 [15]:

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

where subscripts denote partial derivatives, and

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

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

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

where $A = y_u z_v - z_u y_v$, $B = x_v z_u - z_v x_u$ and $C = x_u y_v - y_u x_v$. 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:

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

The mathematical properties of the two surface curvature functions are now discussed in more detail. Both Gaussian and mean curvature values are direction-free quantities. Gaussian and mean curvatures are invariant to arbitrary transformation of the (u, v) parameters as well as rotations and translations of a surface. Combination of these curvature measures enables the local surface type to be categorised. On smoothed surfaces of 3D objects, the procedure for estimating the Gaussian and mean curvatures are as follows. For each point of the surface,

$$p(x(u, v), y(u, v), z(u, v))$$

the corresponding local neighbourhood data is convolved with the partial derivatives of the Gaussian function $G(u, v, \sigma)$. Finally, curvature values on a 3D surface are

estimated by substituting these values into Eqs. (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 that 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. Maxima of torsion 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 . Intuitively, torsion is a local measure of the non-planarity of a space curve [8]. The set of points of a space curve are the values of a continuous, vector-valued, locally one-to-one function

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

where $x(u)$, $y(u)$ and $z(u)$ are the components of $\mathbf{r}(u)$, 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,

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

where $\dot{x}(u)$, $\dot{y}(u)$ and $\dot{z}(u)$ are the convolutions of $x(u)$, $y(u)$ and $z(u)$ with the first derivative of a 1D Gaussian function $G(u, \sigma)$

$$\dot{x} = x * \frac{\partial G}{\partial u}, \quad \dot{y} = y * \frac{\partial G}{\partial u}, \quad \dot{z} = z * \frac{\partial G}{\partial u} \quad (4)$$

* denotes convolution. Note that \ddot{x} and \ddot{y} represent convolutions with the second and third derivatives of G , respectively. While derivative estimation can be sensitive to noise, torsion estimation takes place only after sufficient smoothing has been applied to the data, and is therefore a robust process. This also helps to reduce the number of feature points used for matching later on.

Once Gaussian and mean curvatures have been determined at each point of a 3D surface, zero-crossing contours of those curvatures are recovered from that surface. In general, these contours are space curves. Torsion is computed at each point of those contours using the procedure described above, and the maxima of torsion are then

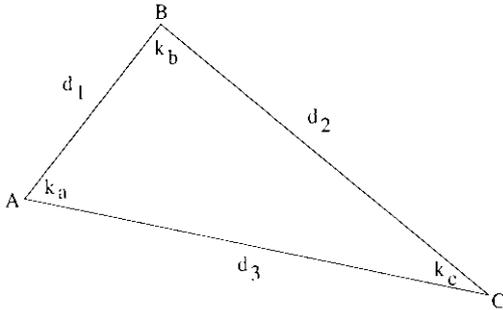


Fig. 1. Triplet of non-collinear points A, B and C.

recovered. These points are added to the set of feature points extracted from the surface.

5. The geometric hashing algorithm

Geometric hashing technique for model-based object recognition was introduced by Lamdan and Wolfson [18,20]. Stein and Medioni [34] as well as Flynn and Jain [7] have also employed geometric hashing for 3D object recognition. In a model-based object recognition system one has to address representation and matching problems. The representation should be rich enough to allow reliable distinction between the different objects in the database as well as efficient matching. A major factor in a reliable representation scheme is its ability to deal with partial occlusion. The objects are represented as sets of geometric features such as points, and their geometric relations are encoded using minimal sets of such features under the allowed object transformations.

5.1. Matching

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 3D objects, the procedure for indexing data into the hash-table is as follows. For each 3D 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 Fig. 1. 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

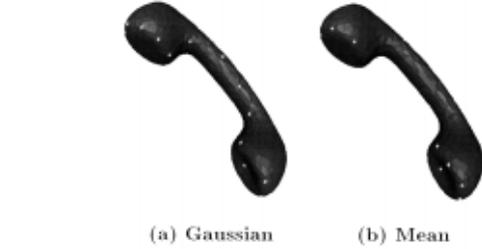


Fig. 2. Curvature maxima of the phone handset.

IV for the hash-table as,

$$IV = \frac{k_a}{k_b} \frac{k_a}{k_c} \frac{d_2 + d_3}{d_1} \tag{5}$$

3. Go back to step (2) and repeat the procedure for different triplets of feature points. All possible triplets of feature points which are not collinear are considered. Note that some of the newly 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 3D 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 3D object can either be complete or incomplete. Thus, given a 3D object in a scene, the matching procedure is as follows:
 4. Repeat steps (1)–(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 a large number of votes close to each other, then the most likely candidate will be chosen using global verification applied at the 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 smoothing process which also removes noise. In order to find the optimal smoothing scale for each object, we employed the following



Fig. 3. Curvature maxima of the dinosaur.

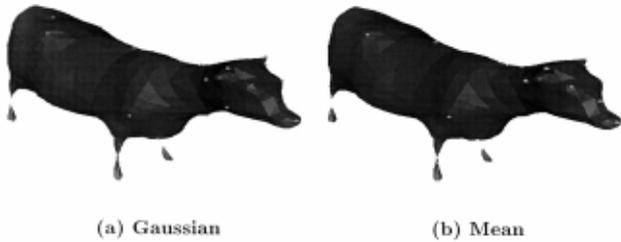


Fig. 4. Curvature maxima of the cow.

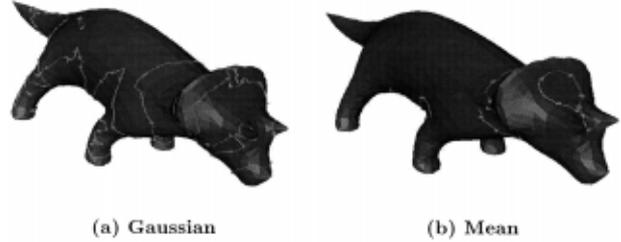


Fig. 6. Torsion maxima of curvature zero-crossing contours of the dinosaur.

multi-scale procedure:

- Apply one iteration of smoothing and count the number of feature points recovered from the object.
- Repeat the first step until several iterations of smoothing have been carried out. Construct a graph of the number of feature points vs. the number of iterations.
- Smooth this graph, and find the point where the slope is minimised. This indicates that the features have become stable at the corresponding scale which is used as the optimal scale for smoothing.

4.2. 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.

Global verification requires the estimation of the 3D transformation parameters for the surviving models. It is possible to make use of closed-form solution techniques [6,10,11,41] to obtain these parameters. However, these techniques are quite complex to implement and relatively inefficient. Considering that the estimation procedure must be repeated many times, it is advantageous to use a method that is as efficient as possible. We have developed a relatively simple and efficient technique which generates approximate solutions. We find that this is quite satisfactory for our application [24].

From the three 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 three 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 three non-collinear points selected from the model object and $P_4(x_{4p}, y_{4p}, z_{4p})$ be a point in the 3D 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 3D 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 [22],

$$\begin{bmatrix} x_{1p} & y_{1p} & z_{1p} & 1 \\ x_{2p} & y_{2p} & z_{2p} & 1 \\ x_{3p} & y_{3p} & z_{3p} & 1 \\ x_{4p} & y_{4p} & z_{4p} & 1 \end{bmatrix} \cdot \begin{bmatrix} a & e & m \\ b & f & n \\ c & g & p \\ d & h & q \end{bmatrix} = \begin{bmatrix} x_{1q} & y_{1q} & z_{1q} \\ x_{2q} & y_{2q} & z_{2q} \\ x_{3q} & y_{3q} & z_{3q} \\ x_{4q} & y_{4q} & z_{4q} \end{bmatrix} \tag{6}$$

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 (6), 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 the x -axis, y -axis and

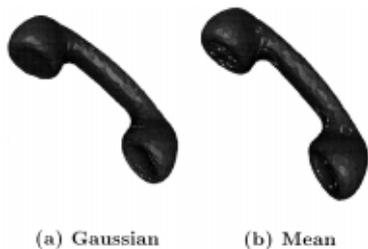


Fig. 5. Torsion maxima of curvature zero-crossing contours of the phone.

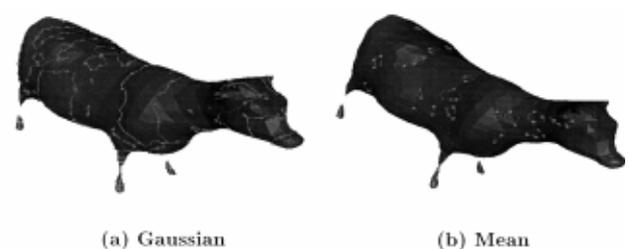


Fig. 7. Torsion maxima of curvature zero-crossing contours of the cow.

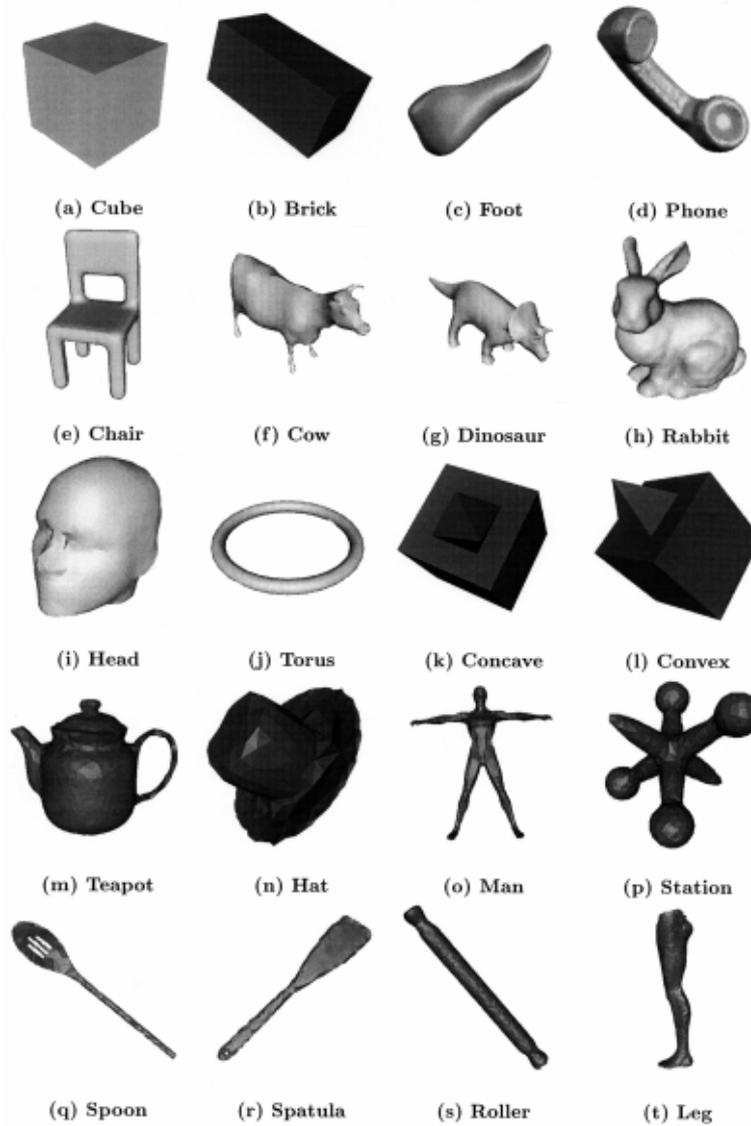


Fig. 8. Free-form 3D objects used for matching experiments.

z -axis denoted $R_x(\gamma)$, $R_y(\beta)$ and $R_z(\alpha)$, respectively, are given by

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (7)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (8)$$

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

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 a 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 Eq. (6),

$$\begin{bmatrix} a \\ e \\ m \\ n \\ p \end{bmatrix} = \begin{bmatrix} \cos \alpha \cos \beta \\ \sin \alpha \cos \beta \\ -\sin \beta \\ \cos \beta \sin \gamma \\ \cos \beta \cos \gamma \end{bmatrix} \quad (10)$$

For the scaling factor \mathfrak{R} , the distances from the centre of gravity points P_4 and Q_4 to their corresponding three points are measured and the shortest distances for each

Table 1
Results of geometric hashing for some rotated/scaled objects

Inputs/models	Chair	Cow	Dinosaur	Foot	Head	Leg	Man	Phone	Rabbit
Chair	68	76	66	1	33	27	52	50	47
Cow	21	85	49	1	19	24	41	29	33
Dinosaur	17	54	83	1	21	21	35	29	32
Foot	20	63	57	91	31	37	45	46	37
Head	21	56	59	2	87	21	39	39	44
Leg	21	72	58	2	20	87	55	50	34
Man	16	48	43	1	17	26	59	30	27
Phone	37	96	90	5	50	65	78	116	65
Rabbit	19	55	53	1	28	21	37	0	87

Table 2
Results of global verification for the rotated/scaled objects

Inputs/models	Chair	Cow	Dinosaur	Foot	Head	Leg	Man	Phone	Rabbit
Chair	357	43	22				19	33	13
Cow									
Dinosaur		277	3665						
Foot		6		35					
Head			24		1309				
Leg		123	28			1029			
Man		224	76				2081	130	
Phone		327	180					9395	
Rabbit									

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 \mathfrak{R} .

$$\mathfrak{R} = \frac{r_1}{r_2}$$

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 objects with the largest clusters are then chosen as the most likely objects present in the scene.

Our clustering algorithm is quite efficient since it avoids the creation of an explicit high-dimensional parameter space. The following is a step-by-step description of our

clustering algorithm:

- Create a cluster for one of the points in the multi-dimensional parameter space. Consider that point as the centre of the cluster.
- Find another point which is closer than a threshold to the centre of the cluster, and add that point to the cluster. If no points are added, go to step 4.
- Compute the new centre of the cluster as the centre of mass of the points already in the cluster. Go to step 2.
- Repeat this procedure for all points which are not already in a cluster.

6. Results and discussion

6.1. Feature extraction

This section presents some results on free-form surface



Fig. 9. Examples of incomplete surfaces.

Table 3
Results of geometric hashing for rotated/scaled incomplete objects

Inputs/models	Chair	Cow	Dinosaur	Foot	Head	Leg	Man	Phone	Rabbit
Chair	70	0	59	1	34	23	41	51	46
Cow	15	46	40	1	18	20	29	32	25
Dinosaur	13	40	68	0	22	14	28	24	30
Foot	25	66	60	38	28	26	49	35	40
Head	19	56	62	0	65	21	54	38	42
Leg	0	0	50	0	0	50	0	0	0
Man	19	0	47	1	24	23	48	34	33
Phone	38	89	87	5	58	56	67	118	68
Rabbit	20	43	46	1	25	21	30	41	52

smoothing and feature extraction. The smoothing routines were implemented entirely in C++. Each iteration of smoothing of a surface with 1000 vertices takes about 0.5 s of CPU time on an UltraSparc 170E. The diffusion and curvature estimation results for 3D surfaces were given [23,44].

It should be noted that Gaussian smoothing does cause shrinkage of objects. In general, this is not a problem unless considered undesirable for a specific application. In fact, we rescale objects after each iteration of smoothing. This procedure cancels out the resulting shrinkage.

Note also that the smoothing process can cause disconnection of object parts. This is a natural possible consequence of the heat diffusion of objects. We apply a decimation procedure after each iteration of smoothing to remove odd-shaped triangles. This procedure can also segment objects into parts when they become very thin. This is due to the fact that very thin regions give rise to odd-shaped (strongly elongated) triangles.

The first example is a phone handset. After smoothing the object, the Gaussian curvatures of all vertices were estimated. Then, the local curvature maxima are computed. The local maxima of Gaussian curvature are displayed on the surface as shown in Fig. 2(a). Fig. 2(b) shows the local maxima of mean curvature for the same object. The local maxima of Gaussian and mean curvatures for the dinosaur and cow are shown in Figs. 3 and 4, respectively. All curvature maxima are shown after one or two iterations.

Next, the torsion maxima of curvature zero-crossing

contours which are alternative features that can be used for matching are determined and displayed on the object. Fig. 5(a) and (b) show the torsion maxima of curvature zero-crossing contours of the phone handset for Gaussian and mean curvatures, respectively. Figs. 6 and 7 show the results for the dinosaur and cow, respectively.

These features are utilised by later processes for robust surface matching and object recognition with occlusion. Animation of surface diffusion can be observed at the web site: <http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/index.html>.

6.2. Matching

This section presents the matching results of the system applied to free-form 3D surfaces in an object database. Given a 3D surface in a scene, the aim is to match the measurements taken at scene points to those memorised in the hash-table. There are 20 different objects in our database. All of these are shown in Fig. 8. It should be pointed out that most of the objects in the database correspond to real range data. They were created by merging range images of real objects obtained from different viewpoints. The matching system was implemented entirely in C++ and ran on an UltraSparc 170E. The system was quite fast with matching times not exceeding 2–3 CPU seconds in each case.

Note that while some object recognition systems have employed greyscale images as their input [12,29], others have made use of range images [5,13] to achieve

Table 4
Results of global verification for rotated/scaled incomplete objects

Inputs/models	Chair	Cow	Dinosaur	Foot	Head	Leg	Man	Phone	Rabbit
Chair	196	11	9				6	19	6
Cow		28	6		5	6	8	17	5
Dinosaur		41	1296						
Foot		7	4	8			4		2
Head		4	4		24		4	6	2
Leg			1			1			
Man		35	17		6	15	370	61	11
Phone		12						781	
Rabbit		8	5		5		13	12	312

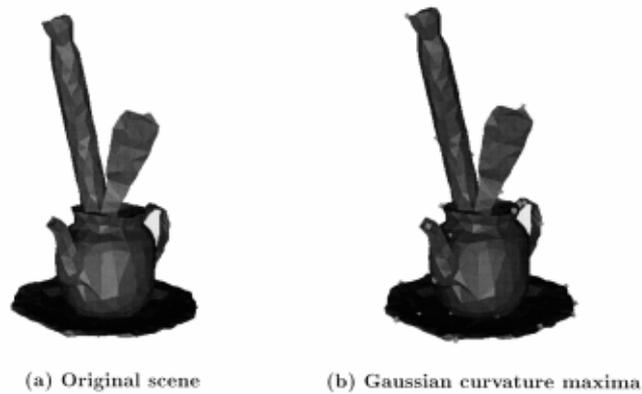


Fig. 10. Kitchen-ware scene and extracted features.

recognition. These systems have been designed for feature extraction and matching based on range images, and cannot cope with more general 3D surfaces. Our system goes further by accepting 3D surfaces that are more general than range images. They can be formed by merging two or more range images obtained from different viewpoints. This makes it possible to obtain more information about the objects to be recognised, and achieve more reliable recognition. Once the local maxima of Gaussian and mean curvatures of each object are obtained, they are then indexed in the hashing table as explained in Section 4.1.

Note that there is no exact correspondence between model vertices and input object vertices. This is because the model and input objects are subjected to different levels of smoothing followed by decimation which modifies the vertex structure on those objects.

The first experiment consisted of applying arbitrary amounts of scaling and 3D rotation to the database objects, and determining 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 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. Table 1 shows the results of geometric hashing for some of the rotated/scaled objects. The numbers shown are the percentages of triangles on each input object which received votes. When the number shown is greater than 100, some triangles received more than one vote. Table 2 shows the results of global verification. The numbers shown are the number of points in the largest clusters for each input object. Blanks indicate that no verification was considered necessary for the corresponding object.

The second experiment made use of incomplete surfaces which were again subjected to arbitrary amounts of scaling and 3D rotation. In order to obtain incomplete surfaces, up to 60% of connected vertices were removed from database objects. The procedure utilised for vertex removal was as follows: a vertex chosen randomly was removed from a given surface. Following that, all of its neighbouring vertices were removed from that surface, and all the neighbours of those vertices, etc. This process continued until the

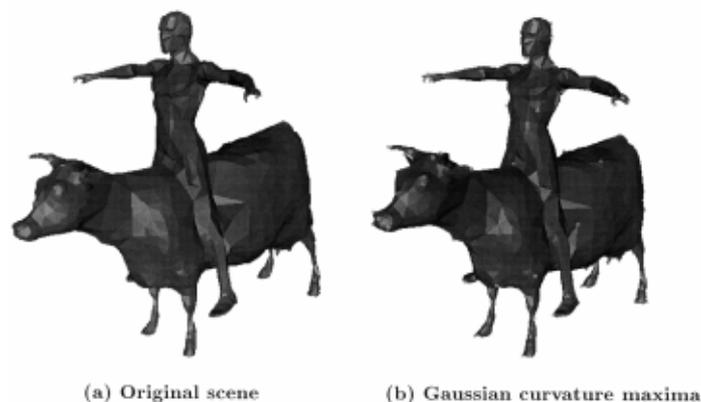


Fig. 11. Bull-rider scene and extracted features.

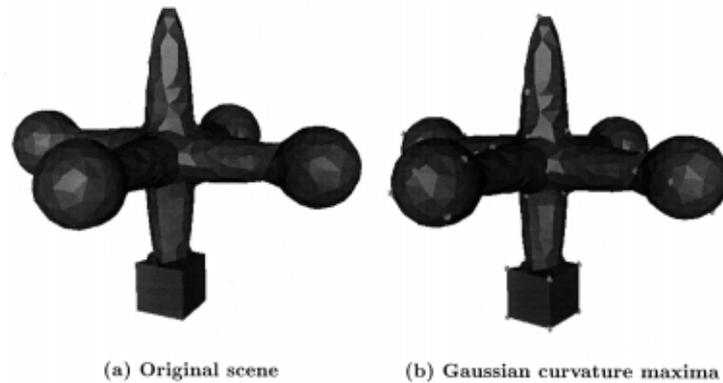


Fig. 12. Space-station scene and extracted features.

desired number of vertices had been removed from the surface. Fig. 9 shows a few examples of incomplete surfaces used in the second experiment.

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. This experiment shows that incomplete surfaces can also be matched successfully to the database models by the system. Table 3 shows the results of geometric hashing for a number of rotated/scaled incomplete objects, and Table 4 shows the results of global verification.

In the third experiment, three complex scenes were created each consisting of two or more objects. Fig. 10 shows the *kitchen-ware* scene and some of the features recovered from that scene. This scene contains a dish, a kettle, a spatula and a roller. Fig. 11 shows the *bull-rider* scene and the corresponding recovered features. This scene contains a cow and a rider. Fig. 12 shows the *space-station* scene and its recovered features. This 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. This experiment shows that scenes depicting occlusion can also be recognised satisfactorily by the system.

7. Conclusions

The recognition of free-form 3D objects using 3D 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 2D Gaussian filters iteratively. Smoothing was used to reduce the 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] P.J. Besl, R.C. Jain, Three dimensional object recognition, *ACM Computing Surveys* 17 (1985) 75–145.
- [2] C. Brechbuhler, G. Gerig, O. Kubler, Parametrization of closed surfaces for 3-d shape description, *Computer Vision and Image Understanding* 61 (2) (1995) 154–170.
- [3] T.W. Chen, W.C. Lin, A neural network approach to CSG-based 3-D object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16 (7) (1994) 719–726.
- [4] R.T. Chin, C.R. Dyer, Model-based recognition in robot vision, *ACM Computing Surveys* 18 (1) (1986) 67–108.
- [5] C.S. Chua, R. Jarvis, Point signatures: a new representation for 3d object recognition, *International Journal of Computer Vision* 25 (1) (1997) 63–85.
- [6] O.D. Faugeras, M. Hebert, The representation, recognition, and locating of 3-D objects, *International Journal of Robotics Research* 5 (3) (1986) 27–52.
- [7] P.J. Flynn, A.K. Jain, Three dimensional object recognition, in: T.Y. Young (Ed.), *Handbook of Pattern Recognition and Image Processing*, vol. 2, 1994, pp. 497–541.
- [8] A. Goetz, *Introduction to Differential Geometry*, Addison-Wesley, Reading, MA, 1970.
- [9] A. Hilton, A.J. Stoddart, J. Illingworth, T. Winder, Reliable surface reconstruction from multiple range images, in *Proceedings of the*

- European Conference on Computer Vision, Cambridge, UK, 1996, pp. 117–126.
- [10] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, *Journal of Optical Society of America A4* (1987) 629–642.
- [11] B.K.P. Horn, H.M. Hilden, S. Negahdaripour, Closed-form solution of absolute orientation using orthonormal matrices, *Journal of Optical Society of America A5* (1988) 1128–1135.
- [12] D.P. Huttenlocher, S. Ullman, Object recognition using alignment, in *Proceedings of the International Conference on Computer Vision*, London, England, 1987, pp. 102–111.
- [13] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (5) (1999) 433–449.
- [14] N. Khalili, F. Mokhtarian, P. Yuen, Free-form surface description in multiple scales: extension to incomplete surfaces, in *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, Slovenia, Ljubljana, 1999, pp. 293–300.
- [15] N. Khalili, F. Mokhtarian, P. Yuen, Recovery of curvature and torsion features from free-form 3-d meshes at multiple scales, in *Proceedings of the Asian Conference on Computer Vision*, Taipei, Taiwan, 2000, pp. 1070–1075.
- [16] J.J. Koenderink, *Solid Shape*, MIT Press, Cambridge, MA, 1990.
- [17] E. Kreyszig, *Differential Geometry*, Oxford University Press, Oxford, 1959.
- [18] Y. Lamdan, J.T. Schwartz, H.J. Wolfson, Object recognition by affine invariant matching, in *Proceedings of CVPR*, Ann Arbor, Michigan, 1988, pp. 335–344.
- [19] Y. Lamdan, J.T. Schwartz, H.J. Wolfson, On recognition of 3-d objects from 2-d images, in *Proceedings IEEE International Conference on Robotics and Automation*, Philadelphia, PA, 1988, pp. 1407–1413.
- [20] Y. Lamdan, H.J. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, in *Proceedings of the International Conference on Computer Vision*, Tampa, FL, 1988, pages 238–249.
- [21] S. Linnainmaa, D. Harwood, L. Davis, Pose determination of a three-dimensional object using triangle pairs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10 (5) (1988) 647–649.
- [22] F. Mokhtarian, A theory of multi-scale, torsion-based shape representation for space curves, *Computer Vision and Image Understanding* 68 (1) (1997) 1–17.
- [23] F. Mokhtarian, N. Khalili, P. Yuen, Multi-scale 3-D free-form surface smoothing, in *Proceedings of the British Machine Vision Conference*, 1998, pp. 730–739.
- [24] F. Mokhtarian, N. Khalili, P. Yuen, Free-form 3-d object recognition at multiple scales, in *Proceedings of the British Machine Vision Conference*, Bristol, 2000.
- [26] A.P. Pentland, Perceptual organisation and the representation of natural form, *Artificial Intelligence* 28 (1986) 293–331.
- [27] M. Pilu, R. Fisher, Recognition of geons by parametric deformable contour models, in *Proceedings of the European Conference on Computer Vision*, Cambridge, UK, 1996, pp. 71–82.
- [28] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
- [29] M. Seibert, A.M. Waxman, Adaptive 3-D object recognition from multiple views, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992) 107–124.
- [30] J.A. Sethian, *Level Set Methods*, Cambridge University Press, Cambridge, 1996.
- [31] S.S. Sinha, R. Jain, Range image analysis, in: T.Y. Young (Ed.), *Handbook of Pattern Recognition and Image Processing: Computer Vision*, vol. 2, 1994, pp. 185–237.
- [32] F. Solina, R. Bajcsy, Recovery of parametric models from range images: the case for superquadrics with global deformations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990) 131–147.
- [33] B.I. Soroka, R.K. Bajcsy, Generalized cylinders from serial selections, in *Proceedings of the International Joint Conference on Pattern Recognition*, 1976, pp. 734–735.
- [34] F. Stein, G. Medioni, Structural indexing: efficient 3-D object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992) 125–145.
- [35] G. Stockman, Object recognition and localization via pose clustering, *Journal of Computer Vision, Graphics and Image Processing* 40 (1987) 361–387.
- [36] A.J. Stoddart, M. Baker, Reconstruction of smooth surfaces with arbitrary topology adaptive splines, in *Proceedings of the European Conference on Computer Vision*, vol. II, 1998, pp. 241–254.
- [37] P. Suetens, P. Fua, A.J. Hanson, Computational strategies for object recognition, *ACM Computing Surveys* 24 (2) (1992) 5–61.
- [38] G. Taubin, Curve and surface smoothing without shrinkage, in *Proceedings of the International Conference on Computer Vision*, 1995, pp. 852–857.
- [39] G. Taubin, T. Zhang, G. Golub, Optimal surface smoothing as filter design, in *Proceedings of the European Conference on Computer Vision*, 1996, pp. 283–292.
- [40] B.M. ter Haar Romeny, *Geometry Driven Diffusion in Computer Vision*, Kluwer Academic, 1994.
- [41] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography — a factorization method, *International Journal of Computer Vision* 9 (2) (1992) 137–154.
- [42] S. Umeyama, Parametrized point pattern matching and its application to recognition of object families, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (2) (1993) 136–144.
- [43] A.K.C. Wong, S.W. Lu, M. Rioux, Recognition and shape synthesis of 3-d objects based on attributed hypergraphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (1989) 279–290.
- [44] P. Yuen, N. Khalili, F. Mokhtarian, Curvature estimation on smoothed 3-d meshes, in *Proceedings of the British Machine Vision Conference*, 1999, pp. 133–142.
- [45] P. Yuen, F. Mokhtarian, N. Khalili, Multi-scale 3-d surface description: open and closed surfaces, in *Scandinavian Conference on Image Analysis*, Greenland, 1999, pp. 303–310.
- [46] P. Yuen, F. Mokhtarian, N. Khalili, and J. Illingworth, Curvature and torsion feature extraction from free-form 3-d meshes at multiple scales, in *IEE Proceedings on Vision, Image and Signal Processing*, 2000.