International Journal of Computer Vision 48(2), 131–149, 2002

Estimation of Error in Curvature Computation on Multi-Scale Free-Form Surfaces

F. MOKHTARIAN, N. KHALILI AND P. YUEN
Centre for Vision, Speech, and Signal Processing, School of Electronic Engineering, Information Technology and Mathematics, University of Surrey, Guildford, GU2 7XH, UK
F.Mokhtarian@ee.surrey.ac.uk
N.Khalili@ee.surrey.ac.uk
P.Yuen@ee.surrey.ac.uk

Received June 2, 1999; Revised October 26, 2000; Accepted October 18, 2001

Abstract. A novel technique for multi-scale curvature computation on a free-form 3-D surface is presented. This is achieved by convolving local parametrisations of the surface with 2-D Gaussian filters iteratively. In our technique, semigeodesic coordinates are constructed at each vertex of the mesh. Smoothing results are shown for 3-D surfaces with different shapes indicating that surface noise is eliminated and surface details are removed gradually. A number of evolution properties of 3-D surfaces are described. Next, the surface Gaussian and mean curvature values are estimated accurately at multiple scales which are then mapped to colours and displayed directly on the surface. The performance of the technique when selecting different directions as an arbitrary direction for the geodesic at each vertex are also presented. The results indicate that the error observed for the estimation of Gaussian and mean curvatures is quite low after only one iteration. Furthermore, as the surface is smoothed iteratively, the error is further reduced. The results also show that the estimation error of Gaussian curvature is less than that of mean curvature. Our experiments demonstrate that estimation of smoothed surface curvatures are very accurate and not affected by the arbitrary direction of the first geodesic line when constructing semigeodesic coordinates. Our technique is independent of the underlying triangulation and is also more efficient than volumetric diffusion techniques since 2-D rather than 3-D convolutions are employed. Finally, the method presented here is a generalisation of the *Curvature Scale Space* method for 2-D contours. The CSS method has outperformed comparable techniques within the MPEG-7 evaluation framework. As a result, it has been selected for inclusion in the MPEG-7 package of standards.

Keywords: free-form surfaces, multi-scale description, local parametrisation, semigeodesic coordinates, Gaussian and mean curvatures, estimation error

1. Introduction

Curvature estimation is an important task in 3-D object description and recognition. Surface curvature provides a unique view-point invariant description of local surface shape. Differential geometry (Goetz, 1970) provides several measures of curvature, which include Gaussian and mean curvatures. Combination of these curvature values enable the local surface type to be categorised.

This paper introduces a new technique for multi-scale curvature computation on a smoothed 3-D surface. Complete triangulated models of 3-D objects are constructed and using a local parametrisation technique, are then smoothed using a 2-D Gaussian filter. The technique considered here is a generalisation of earlier multi-scale representation theories proposed for 2-D contours (Curvature Scale Space method) (Mokhtarian and Mackworth, 1992) and space curves (Torsion Scale Space method) (Mokhtarian, 1997).

The Curvature Scale Space shape descriptor has been selected for MPEG-7 standardization. More details of the diffusion technique as well as a literature survey appear in Mokhtarian et al. (1998).

In our approach, diffusion of the surface is achieved through convolutions of local parametrisations of the surface with a 2-D Gaussian filter (Mokhtarian et al., 1998; Yuen et al., 1999). *Semigeodesic coordinates* (Goetz, 1970) are utilised as a natural and efficient way of locally parametrising surface shape. The most important advantage of our method is that unlike other diffusion techniques such as volumetric diffusion (Koenderink and vanDoorn, 1986; Koenderink, 1990) or level set methods (Sethian, 1996), it has *local support* and is therefore applicable to partial data corresponding to surface-segments. This property makes it suitable for object recognition applications in presence of occlusions. It is also more efficient than those techniques since 2-D rather than 3-D convolutions are employed. Furthermore, note that techniques which apply smoothing in the normal direction need to first estimate curvature in order to displace surface points by a distance proportional to the curvature value. However, curvature estimation itself poses a problem for these methods. Our method overcomes this problem by integrating surface smoothing and curvature estimation into one unified formalism.

Evolution properties of 3-D surfaces are described in this paper. For example, it is shown that our iterative Gaussian filtering converges to the solution of the heat equation. The surface Gaussian and mean curvature values are estimated accurately at multiple scales which are then mapped to colours and displayed directly on the surface provided in the Visualisation Toolkit (VTK) (Schroeder et al., 1996).

The performance of our technique when selecting different directions as an arbitrary direction for the construction of semigeodesic coordinates and hence the diffusion of 3-D surfaces, are also presented. The results indicate that the error observed for the estimation of Gaussian curvature is smaller than that of mean curvature and diffusion reduces estimation error of both surface curvatures.

The organisation of this paper is as follows. Section 2 gives a brief overview of previous work on 3-D object representations including the disadvantage(s) of each method. Section 3 describes the relevant theory from differential geometry and explains how a multiscale shape description can be computed for a free-form 3-D surface. Section 4 contains the evolution properties of 3-D surfaces. Section 5 covers the computation of Gaussian and mean curvatures as well as their error estimations. Section 6 contains a brief summary of an object recognition system based on the technique presented in this paper. Section 7 presents results and discussion. Section 8 contains the concluding remarks.

2. Literature Survey

Many object recognition systems rely on restrictions imposed on the geometry of the object. However, complex free-form surfaces may not be modelled easily using volumetric primitives. A free-form surface is a surface such that the surface normal is defined and continuous everywhere, except at sharp corners and edges (Besl, 1990). Discontinuities in the surface normal or curvature may be present anywhere on a free-form object. The curves that connect these points of discontinuity may meet or diverge smoothly. Recognition of free-form objects is essential in inspection of arbitrary curved surfaces and path planning for robot navigation.

This section presents a survey of previous work on representation of 3-D surfaces. Sinha and Jain (1994) provide an overview of geometry based representations derived from range data of objects. Almost all work on 3-D curvature estimation has been applied to range images rather than 3-D meshes (Berkmann and Caelli, 1994; Flynn and Jain, 1989, 1991; Liang and Todhunter, 1990; Besl and Jain, 1986; Grimson and Lozano-Perez, 1984). Comprehensive surveys of 3-D object recognition systems are presented by Besl and Jain (1985), Chin and Dyer (1986) and Suetens et al. (1992). Some representation schemes for 3-D 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-theoric combinations of volume primitives as in constructive solid geometry (CSG) or spatial occupancy (Pentland, 1986; Solina and Bajcsy, 1990; Chen and Lin, 1994; Samet, 1990). However, it may not be possible to express objects with free-form surfaces using for example, superquadric primitives. Surface-based representations describe an object in terms of the surfaces bounding the object and their properties (Faugeras and Hebert, 1986; Jain and Hoffman, 1988; Flynn and Jain, 1991), and are employed for recognition. 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 (Hilton et al., 1996a, b) and the four sided spline patches. Triangular meshes have been utilised in our work.

Polyhedral approximations (Faugeras and Hebert, 1986) fit a polyhedral object with vertices and relatively large flat faces to a 3-D object. Their disadvantage is that the choice of vertices can be quite arbitrary which renders the representation not robust. Smooth 3-D splines (Stoddart and Baker, 1998) can also be fitted to 3-D 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* (Soroka and Bajcsy, 1976) as well as *geons* (Pilu and Fisher, 1996) approximate a 3-D object using globally parametrised mathematical models, but they are not applicable to detailed free-form objects. *Multi-view* representations (Seibert and Waxman, 1992) are based on a large number of views of a 3-D object obtained from different viewpoints, but difficulties can arise when a non-standard view is encountered. Delingette's work on deformable surfaces (Delingette, 1999; Delingette et al., 1992) is more suitable for surface reconstruction and/or image segmentation. In *volumetric diffusion* (Koenderink, 1990) or level set methods (Sethian, 1996), 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 Taubin (1995) but this method has drawbacks since it is based on weighted averaging using neighbouring vertices and is therefore dependent on the underlying triangulation. The same surface can be triangulated in many different ways. As a result, the outcome of the application of this method will be different in each case. Desbrun et al. (1999) presented a method for 3-D mesh smoothing based on curvature flow, and they devised a way of approximating curvature values on the noisy mesh. However, this gives rise to a so-called *chicken-and-egg* problem. Curvature estimation on a noisy mesh is unreliable. To improve the estimates, the data must be smoothed first. However, their method requires curvature estimates *before* smoothing can be carried out.

The smoothing of 3-D surfaces is a result of the diffusion process (ter Haar Romeny, 1994). For parameterisation of a 3-D surface other methods have also been studied, such as the asymptotic coordinates (Kreyszig, 1959), isothermic coordinates (Goetz, 1970; Chern et al., 1954) and global coordinates (Brechbuhler et al., 1995) used for closed, simply connected objects.

Global representations such as the Extended Gaussian Image (EGI) (Horn, 1984; Kang and Ikeuchi, 1993; Liang and Taubes, 1994) describe 3-D objects in terms of their surface normal distributions on the unit sphere with appropriate support functions. However, arbitrary curved objects have to be either approximated by planar patches or divided into regions based on the Gaussian curvature. Another approach for specifying a 3-D object is the view-centred representations. The graph approach (Koenderink and vanDoorn, 1979) attempts to group a set of infinite 2-D views of a 3-D object into a set of meaningful cluster of appearances. Murase and Nayar (1995) and Swets (1996) also exploit photometric information to describe and recognise objects. A major drawback of view-centred representations is lack of complete information. Part based representations capture structure in object descriptions (Raja and Jain, 1994; Dickinson et al., 1992), but there is a lack of agreement in deciding the general set of part primitives that need to be used in order to be sufficient and appropriate. Furthermore computation of parts from a single view of an object is difficult.

Recent approaches using splash and super polygonal segments (Stein and Medioni, 1992) and algebraic polynomials (Keren et al., 1994; Ponce et al., 1993) have addressed the issue of representing complex curved free-form surfaces. However, there are limitations relating to object segmentation issues, restricting objects to be topologically equivalent to a sphere and sensitivity to noise when low-level surface features are used.

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* (Goetz, 1970). 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 detail in Mokhtarian et al. (1998).

3.1. Geodesic Lines

Geodesic lines are at the heart of a local semigeodesic coordinate system. The following crucial property of geodesic lines is actually utilized to construct geodesics on 3-D triangular meshes:

3.1.1. Minimal Property of Geodesics. An arc of a geodesic line C passing through a point P and lying entirely in a sufficiently small neighborhood of a point P of a surface $\mathcal{S}$ of class C_2 is the shortest join of P with any other point of C by a curve lying in the neighborhood.

3.2. Geodesic Line Construction

To construct semigeodesic coordinates, geodesic lines must be constructed on free-form 3-D meshes. Clearly the segment of a geodesic that lies on any given triangle is a straight line. Two situations must be considered:

- Extension of a geodesic when it intersects a triangle edge
- Extension of a geodesic when it intersects a triangle vertex

Lemma 1 addresses the first situation.

Lemma 1. *Suppose a geodesic intersects an edge e shared by triangles T_1 and T_2. The extension of this geodesic beyond e is obtained by rotating T_2 about e so that it becomes co-planar with T_1, extending the geodesic in a straight line on T_2, and rotating T_2 about e back to its original position.*

Proof: Assume by contradiction that the procedure above does not construct a geodesic. Let g_1 be the segment of the geodesic on T_1 and let g_2 be the segment of the geodesic on T_2. Rotate T_2 about e so that it becomes coplanar with T_1. By assumption, g_1 and g_2 will not be co-linear. Hence, for a point P_1 on g_1 and a point P_2 on g_2, there will be a shorter path from P_1 to P_2. This is the straight line joining P_1 to P_2. Now rotate T_2 back to its original position. The length of the path just constructed remains the same, so it will still be shorter than the geodesic from P_1 to P_2. A contradiction has been reached. Therefore the procedure described correctly constructs a geodesic. Note that the construction above extends to several triangles as long as they remain in a local neighbourhood. □

Lemma 2 addresses the second case.

Lemma 2. *Suppose a geodesic arrives at a vertex V of the mesh. Define the normal vector $\mathbf{n}$ at V as the average of the surface normals of all the triangles incident on V weighted by the incident angle. Let Q be the plane formed by the geodesic incident on V and $\mathbf{n}$. The extension of this geodesic beyond V is found by intersecting Q with the mesh.*

Proof: The curvature vector k of the path obtained by the procedure above lies in Q. k is also perpendicular to tangent plane T (which is defined as perpendicular to $\mathbf{n}$ at V). The vector of geodesic curvature of the path is obtained by projecting k on the tangent plane. It follows that geodesic curvature of the path is zero. Hence the path is a geodesic line. □

Figures 1 and 2 illustrate Lemmas 1 and 2 respectively.

3.3. Semigeodesic Coordinates

Semigeodesic coordinates can be constructed in the following way at a point P on a surface $\mathcal{S}$ of class C_2:

- Choose a geodesic line C through point P in an arbitrary direction.
- Denote by v the arclength parameter on C, such that P corresponds to the value $v = 0$.
- Take further through every point of C the geodesic line L perpendicular to C at the corresponding point.
- Denote by u the arclength parameter on L.

The two parameters u and v determine the position of each point in the domain swept out by these geodesic lines. It can be shown that in a sufficiently small neighborhood of the point P, semigeodesic coordinates can always serve as curvilinear coordinates in a regular

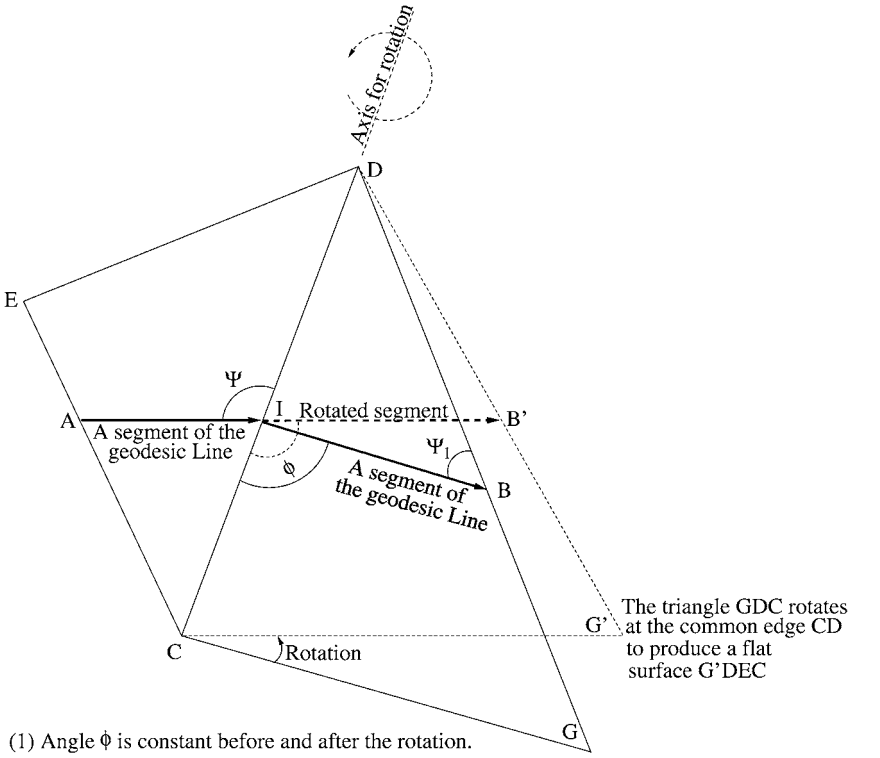


Figure 1. Geodesic line at a triangle edge.

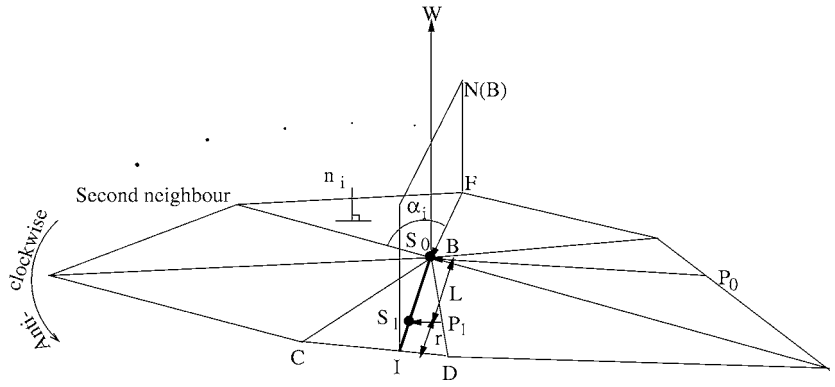


Figure 2. Geodesic line at a vertex.

parametric representation of $\mathcal{S}$ (Goetz, 1970). The orthogonal cartesian coordinates in the plane are a special case of semigeodesic coordinates on a flat surface.

Figure 3 shows the complete semigeodesic coordinates on a triangular mesh.

3.4. Gaussian Filtering of a 3-D Mesh

Gaussian filtering is a weighted average smoothing carried out at a vertex and its neighbourhood. A 2-D Gaussian filter is generated according to the formula

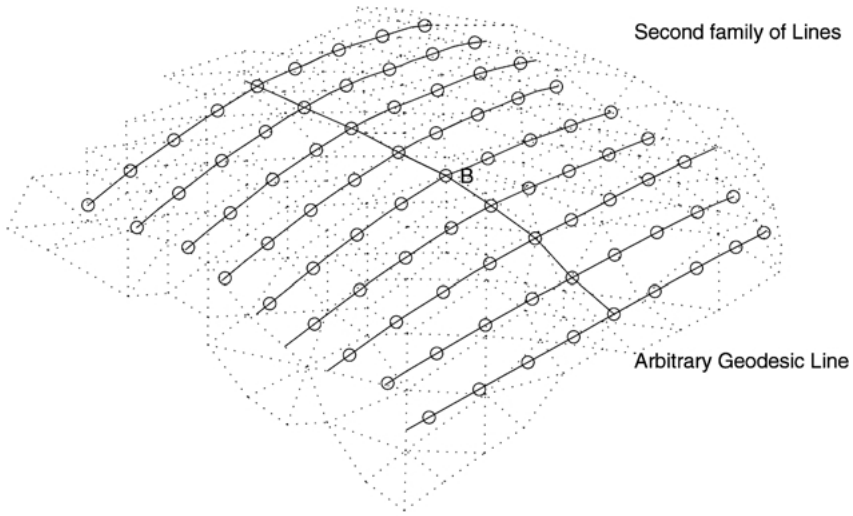


Figure 3. Semigeodesic coordinates on a triangular mesh.

below (Banks, 1990):

$$G(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(u^2+v^2)}{2\sigma^2}} \tag{1}$$

The result of smoothing depends entirely on a vertex and its neighbourhood. In order to minimise over sampling and under sampling, the area size must be neither too large nor too small. In other words, a local area must cover a reasonable size neighbourhood in order to provide accurate results. A local semigeodesic coordinate system may cover a large number of triangles or just a few, depending on the size of those triangles. Experiments were conducted with a filter size of 9 with $\sigma = 1.0$. The unit used for sampling the mesh was the average value of the lengths of all triangle edges on the mesh. The meshes are normalised after each iteration to avoid scale problems.

In order to smooth a 3-D surface, a fixed size 2-D Gaussian filter with $\sigma = 1.0$ is convolved with the local area. Local parametrisation of the surface yields:

$$r(u, v) = (x(u, v),\ y(u, v),\ z(u, v)) \tag{2}$$

The smooth surface is defined by:

$$\mathcal{R}(u, v, \sigma) = (\mathcal{X}(u, v, \sigma),\ \mathcal{Y}(u, v, \sigma),\ \mathcal{Z}(u, v, \sigma)) \tag{3}$$

where

$$\mathcal{X}(u, v, \sigma) = x(u, v) * G(u, v, \sigma)$$
$$\mathcal{Y}(u, v, \sigma) = y(u, v) * G(u, v, \sigma)$$
$$\mathcal{Z}(u, v, \sigma) = z(u, v) * G(u, v, \sigma)$$

and $*$ denotes convolution. This process is repeated at each vertex, and the new vertex positions after filtering define the smoothed surface. This procedure is iterated several times to yield heat diffusion of the surface.

4. Evolution Properties of 3-D Surfaces

Before we present the results on error estimation of curvature computation on 3-D surfaces, the following theorems covering fundamental properties of evolution will be presented in this section.

Theorem 1. *The order of application of evolution and a shape preserving transformation to a surface does not change the final result.*

Proof: Suppose surface $\mathcal{S}$ is evolved into S_σ. Every point of S_σ is a weighted average of a subset of points of $\mathcal{S}$. Therefore evolution at each point Q of $\mathcal{S}$ can be expressed as the convolution of a neighbourhood of Q with a 2-D function (not Gaussian) with unknown

values.

$$P(X, Y, Z) = (x(u, v) \otimes f(u, v), \\ y(u, v) \otimes f(u, v),\ z(u, v) \otimes f(u, v))$$

$\otimes$ denotes convolution. Now apply an affine transform to point P to get $P_1(X_1, Y_1, Y_1)$ where:

$$X_1 = a_1 X + b_1 Y + c_1 Z + d_1$$
$$Y_1 = a_2 X + b_2 Y + c_2 Z + d_1$$
$$Z_1 = a_3 X + b_3 Y + c_3 Z + d_1$$

Alternatively, apply an affine transform to point Q first; and then evolve:

$$X_2 = (a_1 x(u, v) + b_1 y(u, v) \\ + c_1 z(u, v) + d_1) \otimes f(u, v)$$
$$Y_2 = (a_2 x(u, v) + b_2 y(u, v) \\ + c_2 z(u, v) + d_2) \otimes f(u, v)$$
$$Z_2 = (a_3 x(u, v) + b_3 y(u, v) \\ + c_3 z(u, v) + d_3) \otimes f(u, v)$$

so: $X_2 = X_1$, $Y_2 = Y_1$ and $Z_2 = Z_1$. Affine also includes shape preserving transform. □

Theorem 2. *Let $\mathcal{S}$ be a closed surface and let $\mathcal{H}$ be its convex hull. $\mathcal{S}$ remains inside $\mathcal{H}$ during evolution.*

Proof: Since $\mathcal{H}$ is a convex surface, every plane P tangent to $\mathcal{H}$ contains that surface in the left (or right) half-space it creates. Since $\mathcal{S}$ is inside $\mathcal{H}$, $\mathcal{S}$ is also contained in the same half-space. Now rotate P and $\mathcal{S}$ so that P becomes parallel to the xy plane. P is now described by the equation $z = c$. Since P does not intersect $\mathcal{S}$, it follows that $Q_z \geq c$ for every point Q on $\mathcal{S}$. Let S_σ be an evolved version of $\mathcal{S}$. Every point of S_σ is a weighted average of a subset of points of $\mathcal{S}$. Therefore, $R_z \geq c$ for every point R on S_σ, and S_σ is also contained in the same half-space. This result holds for every plane tangent to $\mathcal{H}$; therefore S_σ is contained inside the intersection of all the left (or right) half-spaces created by the tangent planes of $\mathcal{H}$. It follows that S_σ is also inside $\mathcal{H}$. □

Theorem 3. *Iterative Gaussian filtering of a surface converges to the solution of the heat diffusion equation.*

Proof: Let ϵ be the maximum error in the location of any point of surface $\mathcal{S}$ when the heat diffusion of $\mathcal{S}$ is approximated through Gaussian filtering with standard deviation $\Delta\sigma$. Observe that at a point P of $\mathcal{S}$

$$\epsilon = |(\mathbf{r} + Hn) - (\mathbf{r} + \Delta\mathbf{r}_g)| = |Hn - \Delta\mathbf{r}_g|$$

where H is mean curvature, n is the normal vector at P, $\mathbf{r}$ is the position vector of P and $\Delta\mathbf{r}_g$ is the amount of change in the position vector of P after Gaussian filtering. According to heat diffusion equation

$$\frac{\partial \mathbf{r}}{\partial t} = Hn$$

Let

$$\Delta\mathbf{r}_g = H_g n_g$$

where n_g is a unit vector with the same direction as that of $\Delta\mathbf{r}_g$, and H_g is equal to length of $\Delta\mathbf{r}_g$. Let k_1 and k_2 be the principal curvatures at P. Assume that k_1 and k_2 are constant in a small neighborhood of P. The following cases can be distinguished:

- k_1 and k_2 are both zero: the surface is locally planar.
- One of k_1 and k_2 is zero: the surface is locally cylindrical.
- k_1 and k_2 are both positive or both negative: the surface is locally ellipsoidal.
- One of k_1 and k_2 is positive and the other is negative: the surface is locally saddle-shaped.

In each case, it can be confirmed that Gaussian filtering is equivalent to diffusion smoothing of the surface. It follows that for a small $\Delta\sigma$, $H_g \to H$ and $n_g \to n$, and therefore $\epsilon \to 0$. After k iterations of smoothing, total error is given by $k\epsilon$ which is also small. □

Theorem 4. *Let $\mathcal{S}$ be a 3-D surface in C_2. Let S_σ be an evolved version of $\mathcal{S}$ with a cusp point at P. There is a $\delta > 0$ such that $S_{\sigma-\delta}$ intersects itself in a neighborhood of point P.*

Proof: It follows from the equation for heat diffusion

$$\frac{\partial r}{\partial t} = Hn$$

that for two point with infinitesimal distance on $\mathcal{S}$, application of infitesimal diffusion will result in two new points also with infinitesimal distance. This is

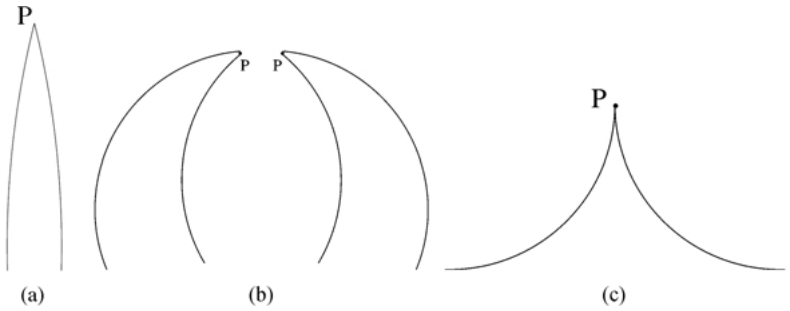


Figure 4. Cross-section of the surface.

because at two nearby points P_1 and P_2, $H_1 \approx H_2$ and $n_1 \approx n_2$ so

$$\frac{\partial r_1}{\partial t} \approx \frac{\partial r_2}{\partial t}$$

It follows further that the tangent planes T_1 and T_2 will also be at infitesimal distance. Now suppose there is a cusp point on S_σ at point P. That cusp point did not exist on the original surface. Consider two points P_1 and P_2 on $S_{\sigma-\delta}$ in a small neighborhood of P. The cusp point at P on S_σ can not be of the forms shown in Fig. 4(a) and (b), since the difference between the tangent planes at P_1 and P_2 on S_σ would be large (which is not possible). It follows that only cusp points of the form shown in Fig. 4(c) are possible since only in this case, the difference between tangent planes at P_1 and P_2 near P is small. Applying reverse diffusion to this object in a neighborhood of P results in a surface that intersects itself near P. It follows that $S_{\sigma-\delta}$ is self-intersecting in a neighborhood of the cusp point. □

Theorem 5. *Simple* (*not self-intersecting*) *surfaces remain simple during evolution.*

Proof: Assume by contradiction that $\mathcal{S}$ is a simple surface that intersects itself during evolution. The location vector of each point of $\mathcal{S}$ is a continuous function of σ during evolution; therefore $\mathcal{S}$ must touch itself at point P before self-intersection. Let S_{σ_0} be such a surface. Consider two neighborhoods S_1 and S_2 of S_{σ_0} that have point P only in common. Hence S_1 and S_2 are non-overlapping. Note that S_1 and S_2 have the same tangent plane at P. Denote this tangent plane by T. The tangent plane exists since it follows from Theorem 4 that P can not be a cusp point on either S_1 or S_2 since S_σ does not self-intersect for $\sigma \leq \sigma_0$. Recall that the infinitesimal movement during arc length evolution of each point of S_1 and S_2 is determined by the equation

$$\frac{\partial \mathbf{r}}{\partial t} = Hn$$

where H is the mean curvature, n is normal and t is time. Therefore during arc length evolution, every point will move in the direction of the normal vector by an amount equal to the curvature at that point. Similarly, during reverse arc length evolution, every point will move in the opposite direction of the normal vector by an amount equal to the curvature at that point. It follows that if S_1 and S_2 are on opposite sides of T, after an infinitesimal amount of reverse arc length evolution they will intersect. This is a contradiction of the assumption that $\mathcal{S}$ was simple before touching itself. Assume then that S_1 and S_2 are on the same side of T. Note that S_1 and S_2 can not be overlapping since they would still be overlapping after an infinitesimal amount of reverse arc length evolution, which is also a contradiction of the assumption that $\mathcal{S}$ was simple before touching itself. Let S_1 be the segment *inside* S_2, i.e., the tangent to S_2 always has S_1 to the same side. It can be seen that S_1 has a larger curvature at P than S_2. Therefore, after an infinitesimal amount of reverse arc length evolution, point P on S_1 and point P on S_2 will move in the same direction, but point P on S_1 will move by a larger amount. It follows that after an infinitesimal amount of reverse arc length evolution, S_1 and S_2 will intersect, which is again a contradiction. It follows that $\mathcal{S}$ remains simple during arc length evolution. □

5. Curvature Estimation

This section presents techniques for accurately estimating 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 (Goetz, 1970). Consider a local parametric representation of a 3-D 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:

$$K = \frac{b_{uu}b_{vv} - b_{uv}^2}{x_v^2 + y_v^2 + z_v^2} \tag{4}$$

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

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

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:

$$H = \frac{b_{vv} + \left(x_v^2 + y_v^2 + z_v^2\right)b_{uu}}{2\left(x_v^2 + y_v^2 + z_v^2\right)} \tag{5}$$

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

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

the corresponding local neighbourhood data is convolved with the partial derivatives of the Gaussian function $G(u, v, \sigma)$, i.e.,

$$x_u = x * \frac{\partial G}{\partial u},\qquad y_u = y * \frac{\partial G}{\partial u},\qquad z_u = z * \frac{\partial G}{\partial u}$$
$$x_v = x * \frac{\partial G}{\partial v},\qquad y_v = y * \frac{\partial G}{\partial v},\qquad z_v = z * \frac{\partial G}{\partial v}$$
$$x_{uu} = x * \frac{\partial^2 G}{\partial u^2},\quad y_{uu} = y * \frac{\partial^2 G}{\partial u^2},\quad z_{uu} = z * \frac{\partial^2 G}{\partial u^2}$$
$$x_{vv} = x * \frac{\partial^2 G}{\partial v^2},\quad y_{vv} = y * \frac{\partial^2 G}{\partial v^2},\quad z_{vv} = z * \frac{\partial^2 G}{\partial v^2}$$
$$x_{uv} = x * \frac{\partial^2 G}{\partial u \partial v},\quad y_{uv} = y * \frac{\partial^2 G}{\partial u \partial v},\quad z_{uv} = z * \frac{\partial^2 G}{\partial u \partial v}$$

where $*$ denotes convolution. Finally, curvature values on a 3-D surface are estimated by substituting these values into Eqs. (4) and (5), respectively.

5.1. *Curvature Error*

For two principal curvatures k_1 and k_2, the Gaussian and mean curvatures are defined as:

$$K = k_1 k_2$$

and

$$H = \frac{k_1 + k_2}{2}$$

Now if $k_1 = k_1 + \varepsilon$ and $k_2 = k_2 + \varepsilon$, where ε represents error ($\varepsilon \ll 1$), then Gaussian curvature is given by,

$$K = k_1 k_2 + \varepsilon(k_1 + k_2) + \varepsilon^2 \tag{6}$$

and mean curvature H is given by

$$H = \frac{k_1 + k_2}{2} + \varepsilon \tag{7}$$

Since k_1 and k_2 are very small values (object sizes are quite large to avoid numerical problems) and

$$\varepsilon(k_1 + k_2) + \varepsilon^2 \ll \varepsilon$$

it follows that for an error in the values of principal curvatures k_1 and k_2, the error introduced in Gaussian curvature is expected to be smaller than that

of mean curvature. Note that Gaussian and mean curvatures have different units. The point of this analysis was simply to demonstrate that the numerical value of the estimation error in Gaussian curvature is smaller than that of mean curvature. On smoothed surfaces of 3-D objects, the procedure for curvature estimation error is as follows:

For each vertex the curvature values are computed for all directions and the average curvature value is then used as the correct value of curvature for that vertex. Then, the error in direction i is given by,

$$error_i = \frac{|\bar{k} - k_i|}{|\bar{k}|} \tag{8}$$

where k_i is the curvature values for direction i, and $\bar{k}$ is the average curvature value.

6. Robust Free-Form 3-D Object Recognition

A system for free-form 3-D object recognition using 3-D models has been developed (Mokhtarian et al., 2000, 2001). The system is based on the technique presented in this paper. Smoothing is utilised to remove noise and to 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 chosen as feature points. Triangles are then formed with feature points as the vertices. Gaussian and mean curvature values are computed at the vertices as well as the lengths of the edges. A *triangle* value defined in terms of vertex curvature values and edge lengths is computed and used to index into a hash table.

High ranking objects from the geometric hashing stage are selected for the global verification stage. The goal of this stage is to ensure that the final selected model is globally consistent with the data. Three-dimensional transform parameters are computed for the local matches detected earlier, and a clustering process is applied. The largest clusters indicate the most likely objects present in the scene. Experiments included 3-D rotation, translation and scaling as well as occlusion and missing data. Our database consisted of 20 objects with both simple and complex shapes. Most of the objects corresponded to real data. Recognition results indicated that the system performed robustly and efficiently.

7. Results and Discussion

This section presents some results on diffusion and curvature estimation as well as the performance of our technique for curvature estimation when *all* possible directions at each vertex on a 3-D surface are selected as an arbitrary direction in order to construct the first geodesic line.

7.1. Diffusion

The smoothing routines were implemented entirely in C++. Complete triangulated models of 3-D objects used for our experiments are constructed at our centre (Hilton et al., 1996b). Each iteration of smoothing of a surface with 1000 vertices takes about 0.5 second of CPU time on an UltraSparc 170E. Smoothing results are shown where the direction of one of the incident edges is selected as an arbitrary direction and the surface is also sampled locally at a step size equal to 1000.

Note that without postprocessing our smoothing method eventually shrinks objects towards their centre of mass. This shrinking effect is not a problem unless it is considered undesirable in a specific application. In fact, we cancel out the shrinking by rescaling objects after each iteration of smoothing.

The first test object was a foot with 2898 triangles and 1451 vertices, which becomes rounded iteratively and evolves into an ellipsoidal shape after 100 iterations as shown in Fig. 5. The second test object was a rabbit with 1996 triangles and 1000 vertices as shown in Fig. 6. The ears disappear after 10 iterations and the object becomes a smooth and rounded shape. The third test object was a dinosaur with 2996 triangles and 1500 vertices as shown in Fig. 7. The object becomes smoother gradually and the legs, tail and ears are removed after 17 iterations. The fourth test object was a cow with 3348 triangles and 1676 vertices as shown in Fig. 8. The surface noise is eliminated iteratively with the object becoming smoother gradually where after 15 iterations the legs, ears and tail are removed, as was seen for the dinosaur. The final test object was a chair with 3788 triangles and 1894 vertices as shown in Fig. 9. This is a different type of object with a hole. As can be seen, holes do not casue any problems for this method.

7.2. Curvature Estimation

This section presents the results of application of our curvature estimation techniques to 3-D objects using

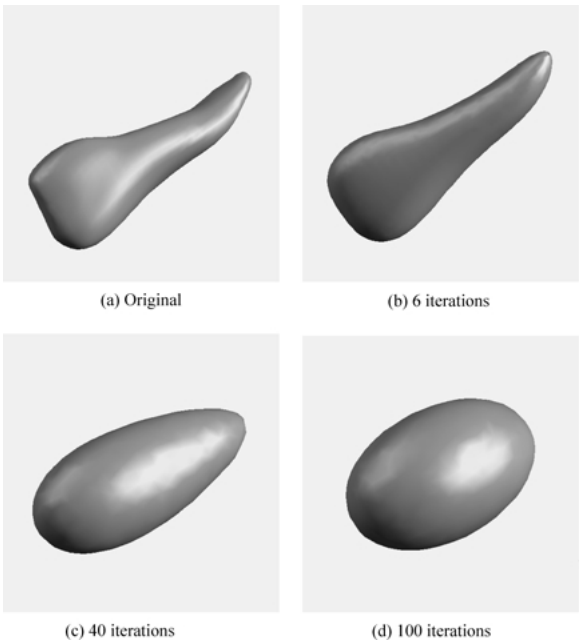


Figure 5. Diffusion of the foot.

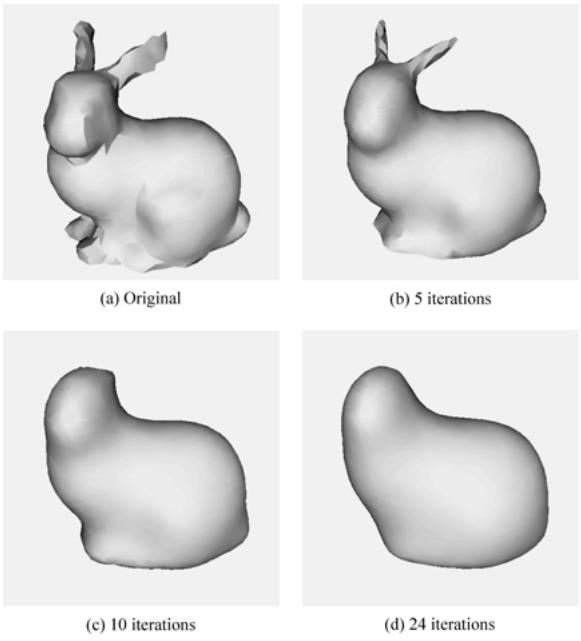


Figure 6. Diffusion of the rabbit.

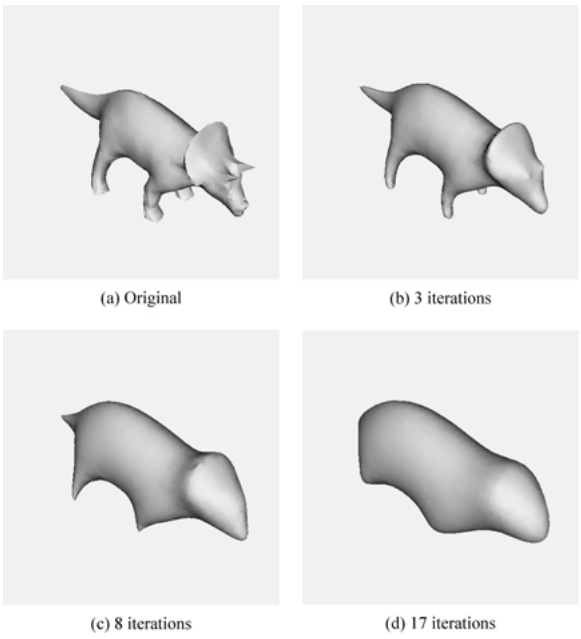


Figure 7. Smoothing of the dinosaur.

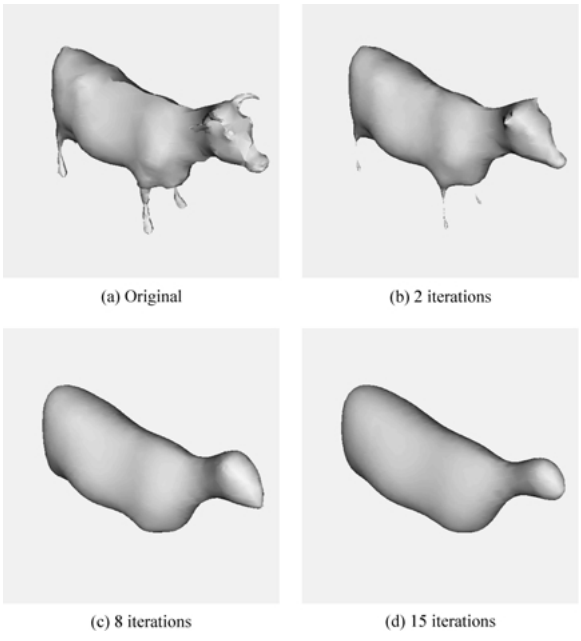


Figure 8. Smoothing of the cow.

methods described in Section 5. The diffusion results for other 3-D surfaces were given in Mokhtarian et al, (1998). The first example is a foot. After smoothing the object, the Gaussian curvatures of all vertices were estimated using Eq. (4). To visualise these curvature values on the surface, they are then mapped to colours using the Visualisation Toolkit (VTK) (Schroeder et al., 1996), and the results are shown in Fig. 10(a). Surface curvature colours are coded as follows: *red = high, blue = low and other colours designate in-between*

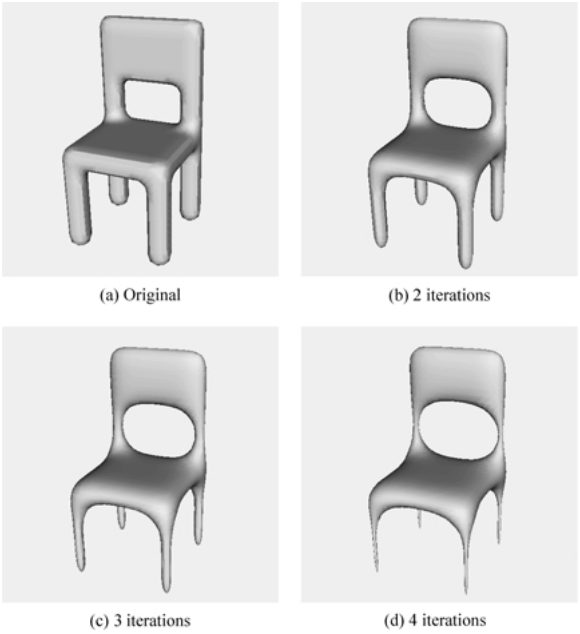


Figure 9. Diffusion of the chair.

values. All convex corners of the foot are red, indicating high curvature values, whereas the concave corners are blue indicating low curvature values and flat areas are green since their curvature values are close to zero. The same experiment was repeated to estimate the mean curvatures of the foot using Eq. (5) and the results are shown in Fig. 10(b). This indicates that mean curvature values for the edges are different than those for flat areas, as expected.

The next object was a rabbit. Its Gaussian curvature values were estimated and results are shown in Fig. 11(a). These results again confirm that the curvature values are high and low at convex and concave corners, respectively. The mean curvatures of the rabbit were also estimated and the results are shown in Fig. 11(b). Gaussian and mean curvatures were also estimated for more complex objects. Figures 12 and 13 show the results for a dinosaur and a cow, respectively.

7.3. Estimation of Error in Curvature Computation

We first applied our curvature estimation technique to a surface with known curvature values. Our method was tested on a 3-D mesh representing a sphere. It was confirmed that Gaussian and mean curvature values estimated at the vertices were approximately equal. The errors observed were in agreement with expectations (see below). However, we believe that the use of objects

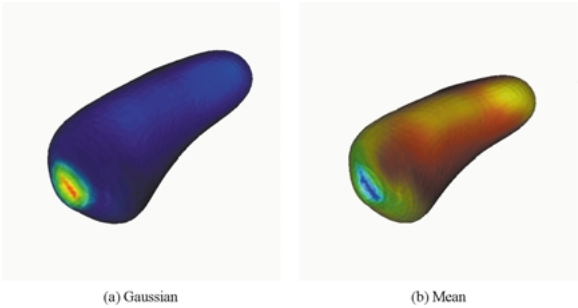


Figure 10. Gaussian and mean curvatures on the foot.

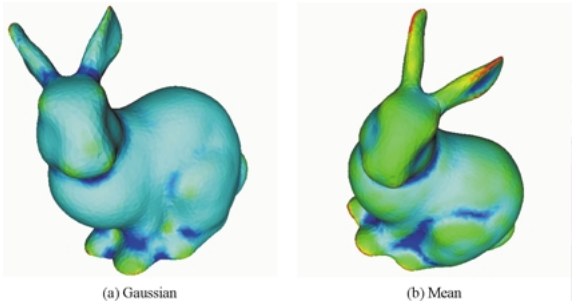


Figure 11. Gaussian and mean curvatures on the rabbit.

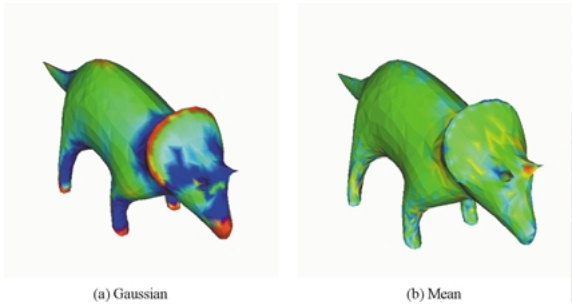


Figure 12. Gaussian and mean curvatures on the dinosaur.

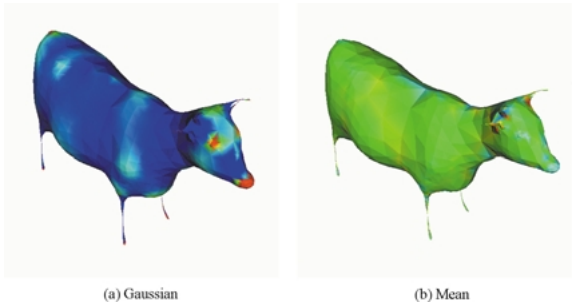
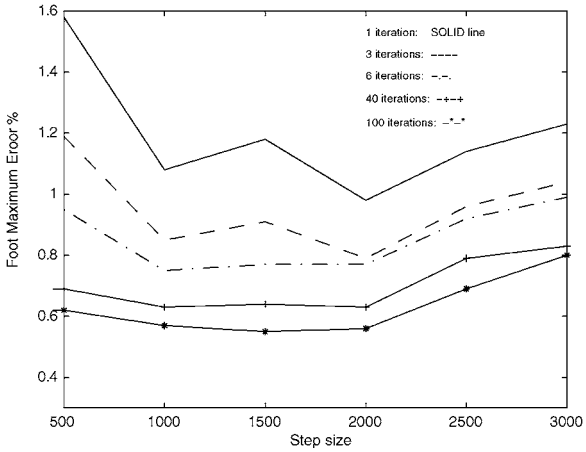


Figure 13. Gaussian and mean curvatures on the cow.

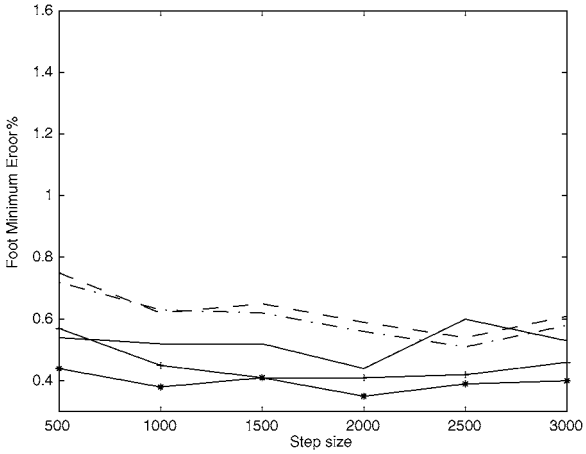
with known curvatures is not a satisfactory method of testing our method. Our technique was designed for curvature estimation on noisy, free-form meshes with unknown curvature values. Objects with known curvatures tend to have quite simple shapes. As a result, it is not difficult to devise techniques for curvature estimation on those objects which will perform well by exploiting constraints about object shape.

As mentioned before, in our method the direction of the first geodesic line is randomly selected. We will therefore examine curvature estimation error for *all* possible directions. The smoothing procedure was repeated where for each vertex all possible directions for the first geodesic line were constructed and different step sizes were considered. Specifically, the direction of each incident edge was defined as a possible direction. After the object was smoothed, for each vertex the curvature values were computed for all directions and the curvature errors were estimated using Eq. (8). Then, the maximum, minimum and average values of the curvature errors were computed. Figure 14(a) shows the error distribution for estimating maximum Gaussian curvature with the step size varying from 500 to 3000. These results indicate that for the step size between 1000 to 2000 the error is reduced to about 1.0% after one iteration. We then repeated the experiment when for each vertex the minimum error in curvature values of all possible directions were computed and results are shown in Fig. 14(b). These indicate that for the step size between 1000 to 2000 the error is reduced to about 0.5% after one iteration. When average curvature value of all possible directions was calculated the average curvature error is about 0.75% after one iteration, as shown in Fig. 14(c). However, as the surface becomes smooth iteratively, the errors reduce as shown in Fig. 14. After 100 iterations the errors in maximum, minimum and average curvature values are reduced to 0.55%, 0.35% and 0.45%, respectively.

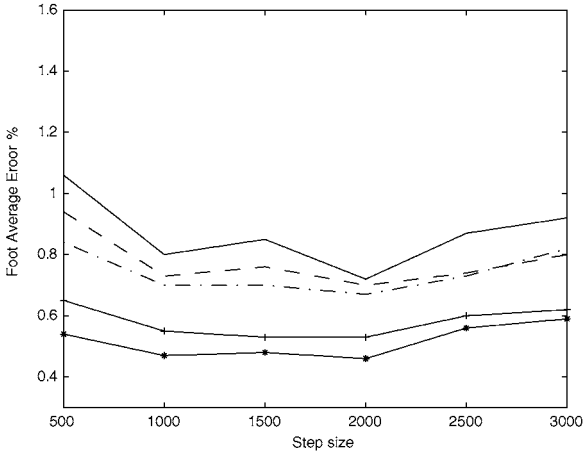
The above procedures were also repeated for estimation of mean curvatures. Figure 15(a) and (b) shows the error distributions for the estimation of mean curvatures, and the error is also reduced for the step sizes between 1000 to 2000, which are about 3.0% and 1.0% for the maximum and minimum mean curvatures, respectively. For the average value of mean curvature, the error is about 2.0% as shown in Fig. 15(c). As the surface becomes smooth iteratively, the errors are reduced as shown in Fig. 15 and after 100 iterations the errors in maximum, minimum and average curvature values drop to about 2.2%, 0.5% and 1.4%, respectively.



(a) Maximaum Error

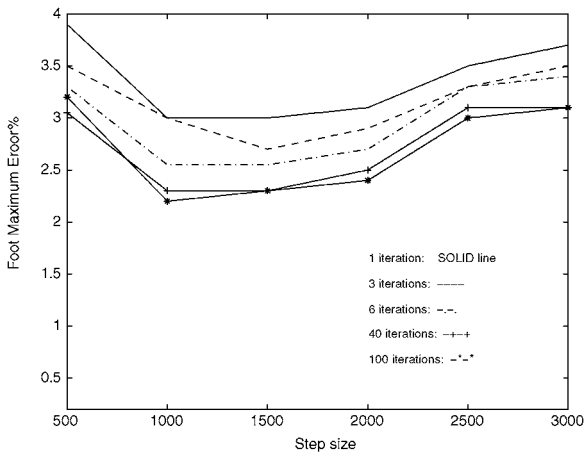


(b) Minimum Error

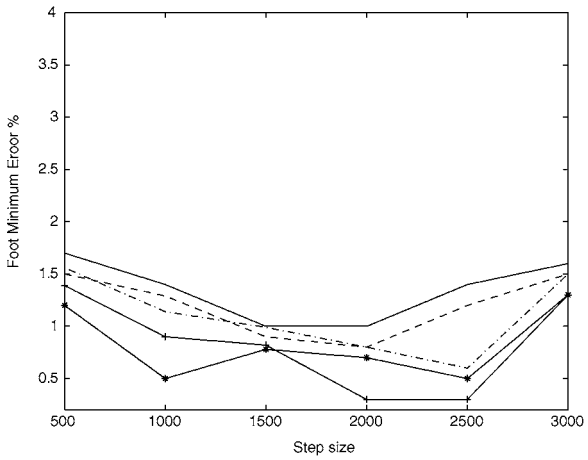


(c) Average Error

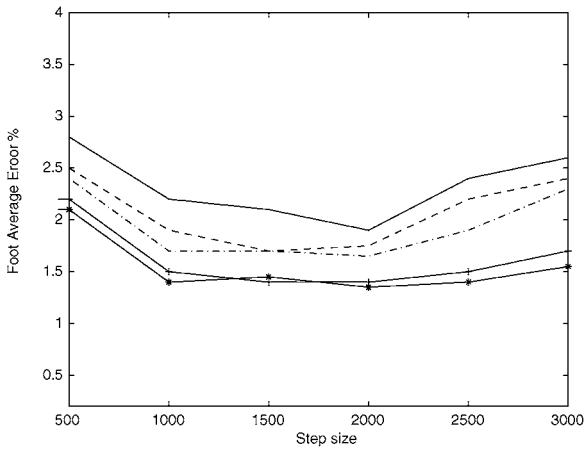
Figure 14. Gaussian curvature error distribution of the foot.



(a) Maximum Error

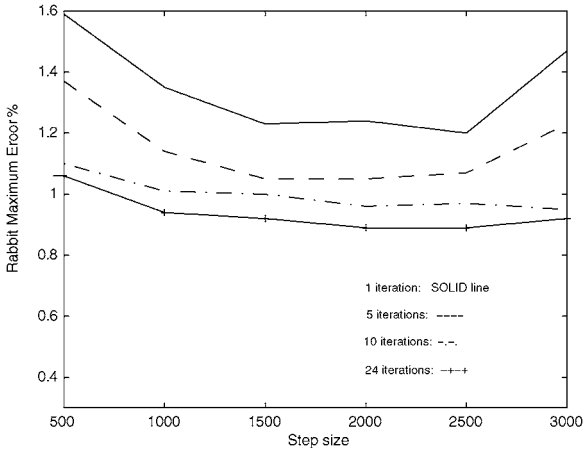


(b) Minimum Error

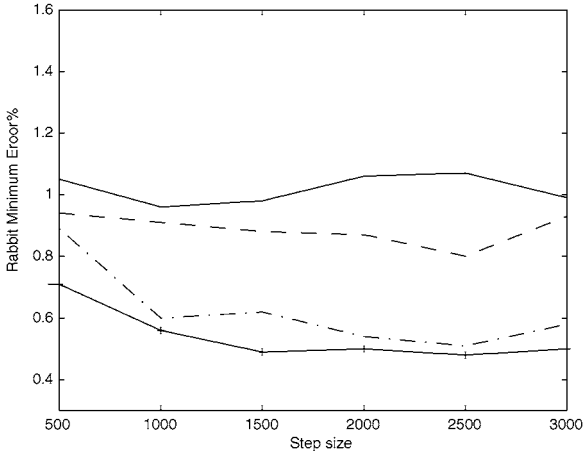


(c) Average Error

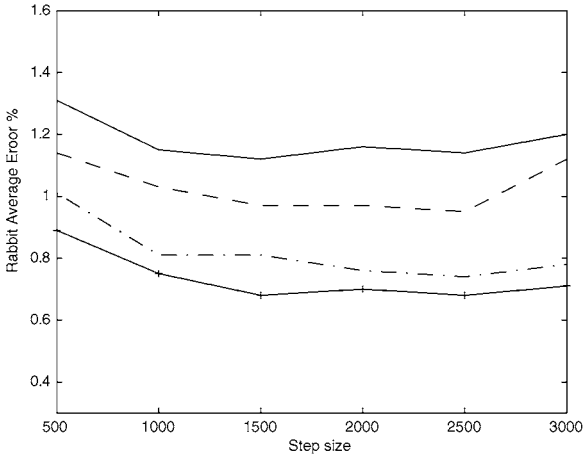
Figure 15. Mean curvature error distribution of the foot.



(a) Maximum Error

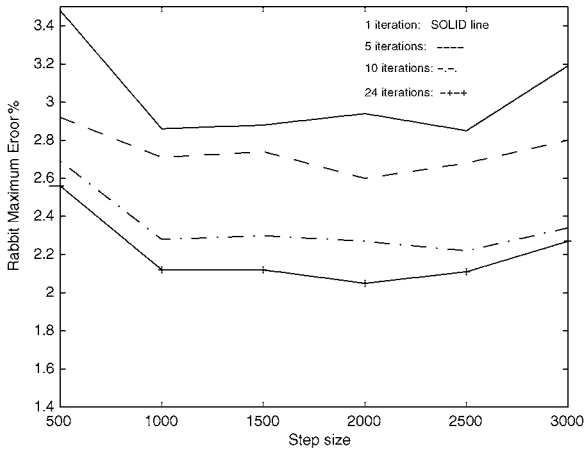


(b) Minimum Error

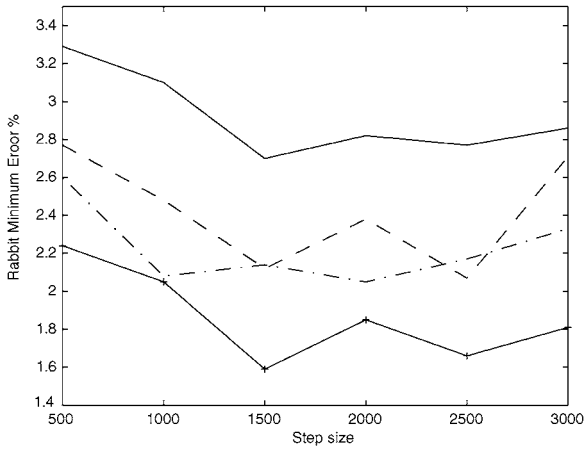


(c) Average Error

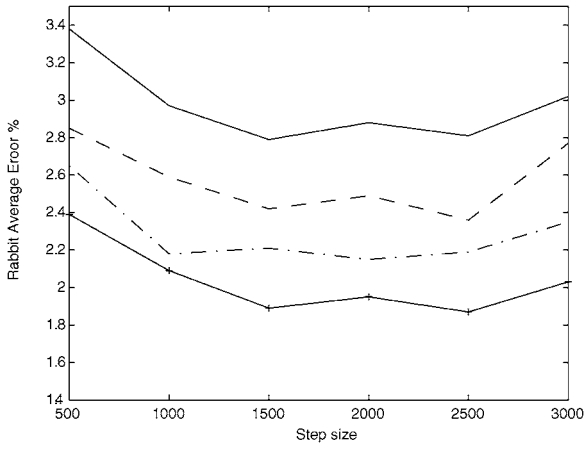
Figure 16. Gaussian curvature error distribution of the rabbit.



(a) Maximum Error

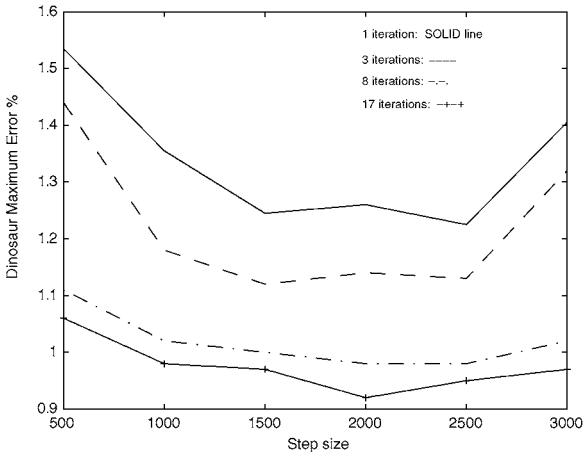


(b) Minimum Error

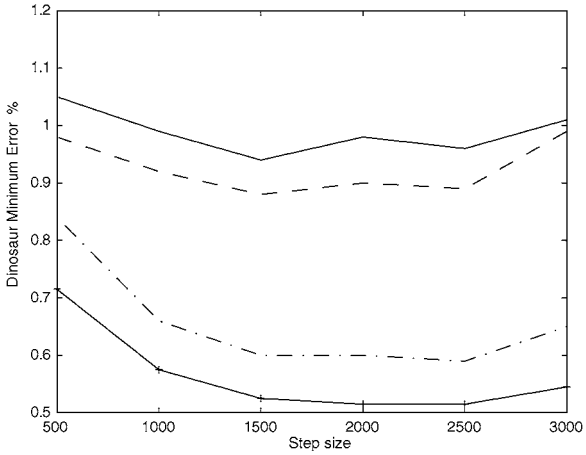


(c) Average Error

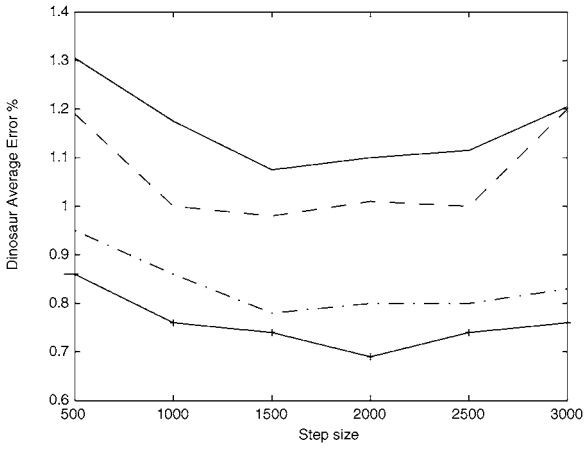
Figure 17. Mean curvature error distribution of the rabbit.



(a) Maximum Error

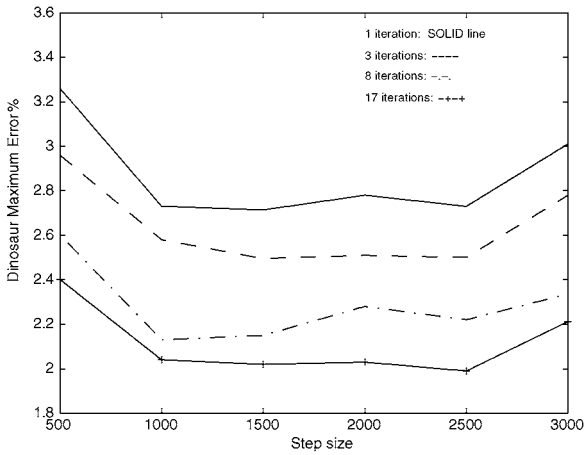


(b) Minimum Error

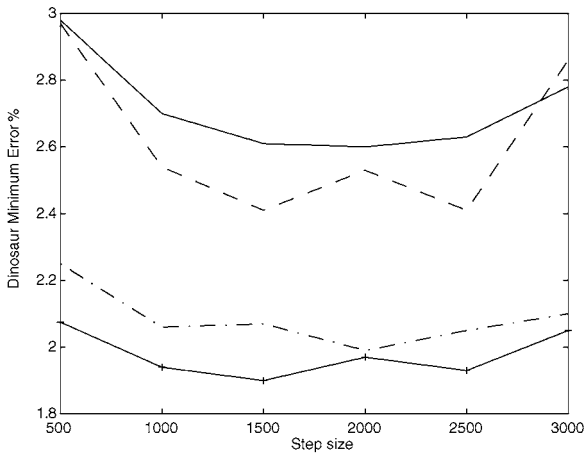


(c) Average Error

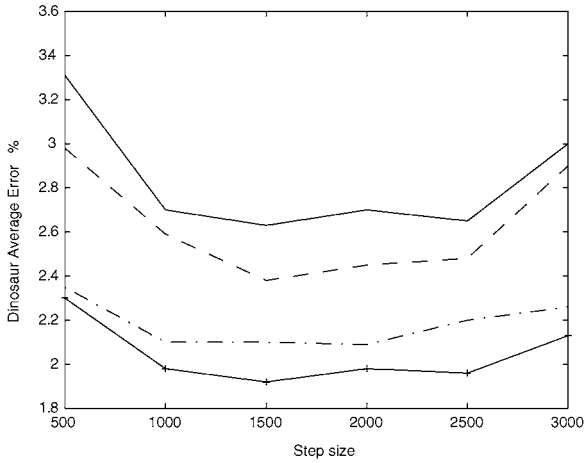
Figure 18. Gaussian curvature error distribution of the dinosaur.



(a) Maximum Error

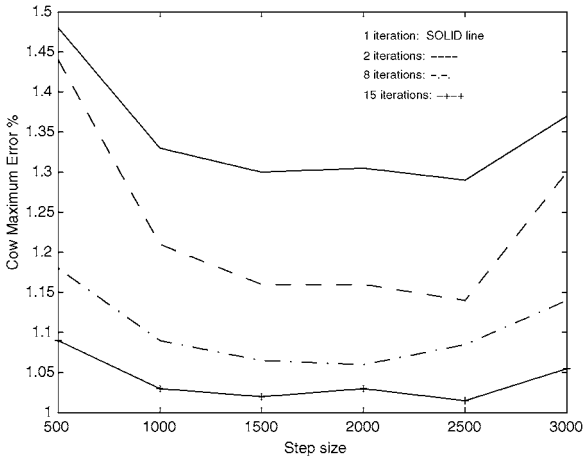


(b) Minimum Error

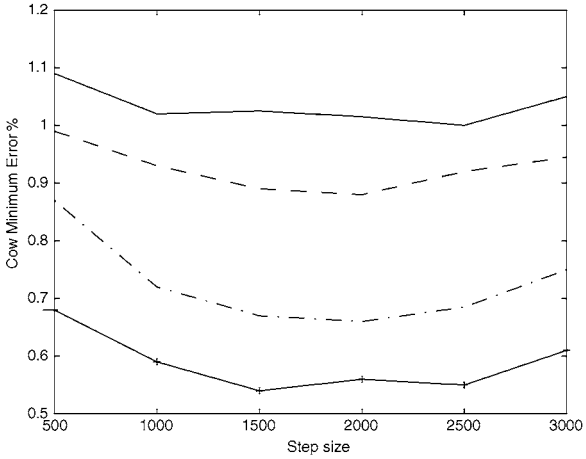


(c) Average Error

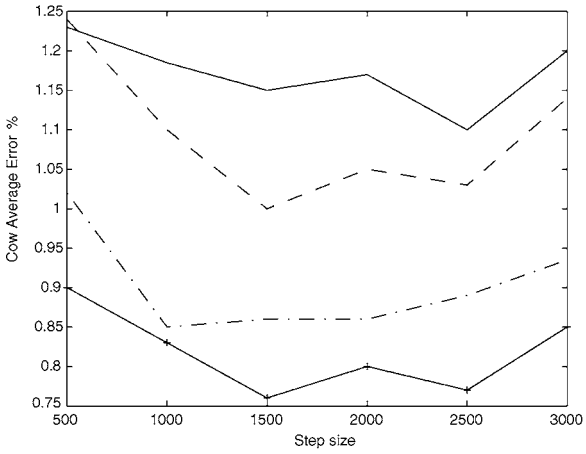
Figure 19. Mean curvature error distribution of the dinosaur.



(a) Maximum Error



(b) Minimum Error



(c) Average Error

Figure 20. Gaussian curvature error distribution of the cow.

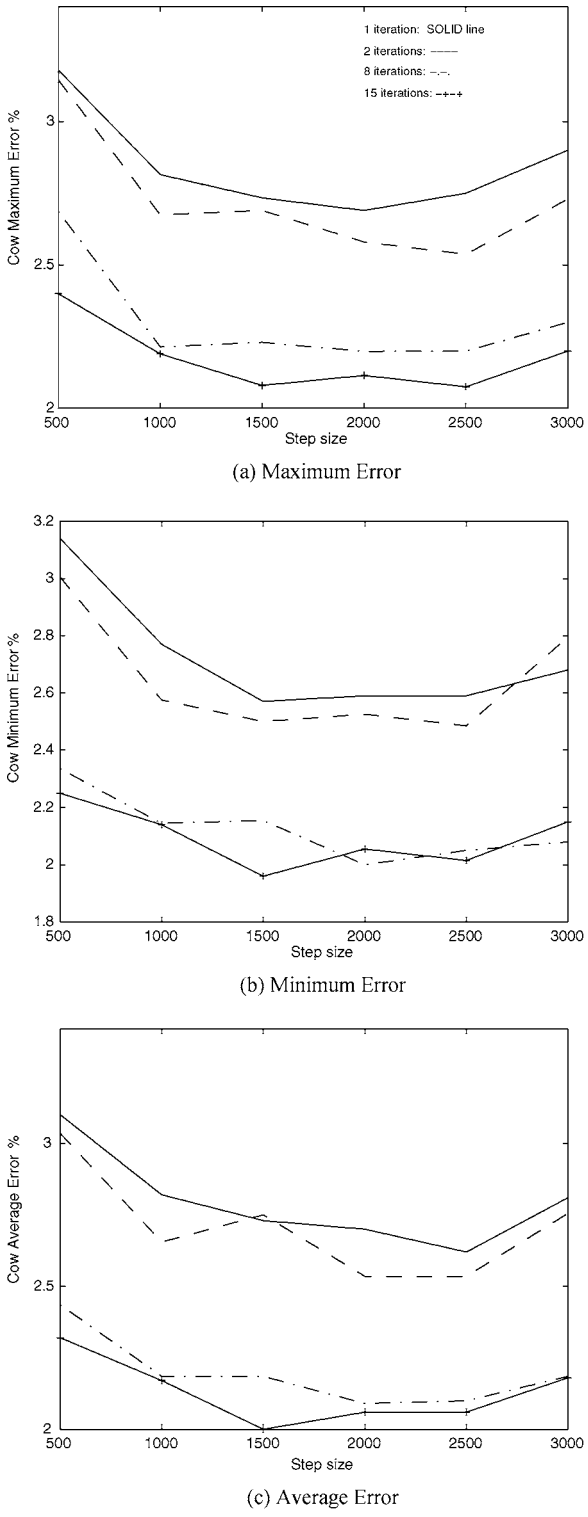


Figure 21. Mean curvature error distribution of the cow.

Notice that the error for the estimation of Gaussian curvatures are lower than that of the mean curvature as was discussed in the previous section. Figure 16 shows the error distribution for estimating Gaussian curvatures of the rabbit when all possible directions were selected. Again the errors were reduced for step sizes between 1000 to 2000 and for one iteration, the errors for maximum, minimum and average Gaussian curvatures are about 1.23%, 0.96% and 1.15%, respectively. After 24 iterations these errors reduce to about 0.92%, 0.5% and 0.7%, respectively. Figure 17 also shows the error distribution for estimating mean curvature of the rabbit and these results also indicate that the errors for maximum, minimum and average curvature values are about 2.85%, 2.7% and 2.8%, respectively. After 24 iterations these errors reduce to about 2.1%, 1.6% and 1.9%, respectively.

Figure 18 shows the error distribution for estimating Gaussian curvatures of the dinosaur when all possible directions were selected. The errors were reduced for step sizes between 1000 to 2500 and for one iteration, the errors for maximum, minimum and average Gaussian curvatures are about 0.94%, 1.23% and 1.10%, respectively. After 17 iterations these errors reduce to about 0.52%, 0.92% and 0.7%, respectively. Figure 19 also shows the error distribution for estimating mean curvature of the dinosaur and these results also indicate that the errors for maximum, minimum and average curvature values are about 2.60%, 2.72% and 2.65%, respectively. After 17 iterations these errors reduce to about 2.0%, 1.9% and 1.92%, respectively. Figure 20 shows the error distribution for estimating Gaussian curvatures of the cow when all possible directions were selected. Again the errors were reduced for step sizes between 1000 to 2000 and for one iteration, the errors for maximum, minimum and average Gaussian curvatures are about 1.29%, 1.0% and 1.1%, respectively. After 15 iterations these errors reduce to about 1.0%, 0.54% and 0.76%, respectively. Figure 21 further shows the error distribution for estimating mean curvature of the cow and these results also indicate that the errors for maximum, minimum and average curvature values are about 2.70%, 2.57% and 2.62%, respectively. After 15 iterations these errors reduce to about 2.07%, 1.96% and 2.0%, respectively.

Our experiments indicate that estimation of Gaussian and mean curvatures on smoothed surfaces is very accurate and not affected by the arbitrary direction of the first geodesic line when constructing

semigeodesic coordinates. Smoothing is necessary to remove noise from the surface before curvature can be estimated reliably. While smoothing might cause a displacement of features, since scale changes slowly, the resulting displacement is small, and therefore, it is not difficult to determine the correspondence between the features across scales. Our technique was also applied to a number of incomplete surfaces with the results shown in Yuen et al. (1999). Animation of surface diffusion can be observed at: *http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/index.html*

8. Conclusions

A novel technique for multi-scale curvature computation on a free-form 3-D surface is presented. Semigeodesic coordinates were constructed at each vertex of the mesh which became the local origin. A geodesic from the origin was first constructed in an arbitrary direction such as the direction of one of the incident edges. During the diffusion process, 3-D surfaces were also sampled locally using different step sizes. Complete triangulated models of 3-D objects were constructed and using a local parametrisation technique, were then smoothed using a 2-D Gaussian filter. The smoothing eliminated the surface noise and small surface detail gradually, and resulted in gradual simplification of object shape. Evolution properties of 3-D surfaces were described in this paper. For example, it was shown that our iterative Gaussian filtering converges to the solution of the heat equation. The performance of our technique for curvature estimation when selecting different directions as an arbitrary direction at each vertex was also presented and results indicated that the errors involved in the estimation of Gaussian and mean curvatures were quite small after only one iteration. Furthermore, as the surface became smoothed iteratively, the error was further reduced for both Gaussian and mean curvatures. These results indicated that estimation of Gaussian and mean curvatures on smoothed surfaces is very accurate and not affected by the arbitrary direction of the first geodesic line when constructing semigeodesic coordinates. It was also shown that the error for the estimation of Gaussian curvature is less than that of the mean curvature. The surface Gaussian and mean curvatures were also mapped to colours, and shown directly on the surface provided in the Visualisation Toolkit (VTK). All convex corners of the surface indicated high Gaussian curvature values, whereas the concave corners indicated low Gaussian curvature values and the curvature values of flat areas are close to zero.

References

Banks, S.P. 1990. *Signal Processing, Image Processing and Pattern Recognition*. Prentice-Hall: Englewood Cliffs, NJ.

Berkmann, J. and Caelli, T. 1994. Computation of surface geometry and segmentation using covariance techniques. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(11):1114–1116.

Besl, P.J. 1990. The free-form surface matching problem. In *Machine Vision for Three-Dimensional Scenes*, H. Freeman (Ed.), pp. 25–71.

Besl, P.J. and Jain, R.C. 1985. Three dimentional object recognition. *ACM Computing Surveys*, 17:75–145.

Besl, P.J. and Jain, R.C. 1986. Invariant surface characteristics for 3D object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33:33–80.

Brechbuhler, C., Gerig, G., and Kubler, O. 1995. Parametrization of closed surfaces for 3-D shape description. *Computer Vision and Image Understanding*, 61(2):154–170.

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

Chern, S.S., Hartman, P., and Wintner, A. 1954. On isometric coordinates. In *Commentaries Mathematical Helvetici*, vol. 28.

Chin, R.T. and Dyer, C.R. 1986. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108.

Delingette, H. 1999. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146.

Delingette, H., Hebert, M., and Ikeuchi, K. 1992. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–140.

Desbrun, M., Meyer, M., Schroder, P., and Barr, A.H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH*, Los Angeles, pp. 317–324.

Dickinson, S.J., Pentland, A.P., and Rosenfeld, A. 1992. 3-D shape recovery using distributed aspect matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:174–198.

Faugeras, O.D. and Hebert, M. 1986. The representation, recognition, and locating of 3-D objects. *International Journal of Robotics Research*, 5(3):27–52.

Flynn, P.J. and Jain, A.K. 1989. On reliable curvature estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 110–116.

Flynn, P.J. and Jain, A.K. 1991. Bonsai: 3-D object recognition using constrained search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10:1066–1075.

Goetz, A. 1970. *Introduction to Differential Geometry*. Addison-Wesley: Reading, MA.

Grimson, W.E.L. and Lozano-Perez, T. 1984. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35.

Hilton, A., Stoddart, A.J., Illingworth, J., and Windeatt, T. 1996a. Marching triangles: Range image fusion for complex object modelling. In *Proc. IEEE International Conference on Image Processing*, Lausanne, Switzerland, pp. 381–384.

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

Horn, B.K.P. 1984. Extended gaussian images. In *Proc. of the IEEE*, 72(12).

Jain, A.K. and Hoffman, R.L. 1988. Evidence based recognition of 3-D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10:783–802.

Kang, S.B. and Ikeuchi, K. 1993. The complex egi, a new representation for 3-D pose determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(7):707–721.

Keren, D., Cooper, D., and Subrahmonia, J. 1994. Describing complicated objects by implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:38–53.

Koenderink, J.J. 1990. *Solid Shape*. MIT Press: Cambridge, MA.

Koenderink, J.J. and vanDoorn, A.J. 1979. Internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32(4):211–216.

Koenderink, J.J. and vanDoorn, A.J. 1986. Dynamic shape. *Biological Cybernetics*, 53:383–396.

Kreyszig, E. 1959. *Differential Geometry*. Oxford University Press: Oxford, UK.

Liang, P. and Taubes, C.H. 1994. Orientation-based differential geometric representations for computer vision applications. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(3):249–258.

Liang, P. and Todhunter, J.S. 1990. Representation and recognition of surface shapes in range images: A differential geometry approach. *Computer Vision, Graphics, and Image Processing*, 52(1):78–109.

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

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

Mokhtarian, F., Khalili, N., and Yuen, P. 2000. Free-form 3-D object recognition at multiple scales. In *Proc. British Machine Vision Conference*, Bristol, pp. 446–455.

Mokhtarian, F., Khalili, N., and Yuen, P. 2001. Multi-scale free-form 3D object recognition using 3D models. *Image and Vision Computing*, 19:271–281.

Mokhtarian, F. and Mackworth, A.K. 1992. A theory of multi-scale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(8):789–805.

Murase, H. and Nayar, S.K. 1995. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24.

Pentland, A.P. 1986. Perceptual organisation and the representation of natural form. *Artificial Intelligence*, 28:293–331.

Pilu, M. and Fisher, R. 1996. Recognition of geons by parametric deformable contour models. In *Proc. European Conference on Computer Vision*, Cambridge, UK, pp. 71–82.

Ponce, J., Kriegman, D.J., Petitjaan, S., Sullivan, S., Taubin, G., and Vijayakumar, B. 1993. Representations and algorithms for 3-D curved object recognition. In *Three-Dimensional Object Recognition Systems*, Amsterdam, The Netherlands, pp. 17–56.

Raja, N.S. and Jain, A.K. 1994. Obtaining generic parts from range images using a multi-view representation. *Computer Vision, Graphics and Image Processing*, 60:44–64.

Samet, H. 1990. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley: Cambridge, MA.

Schroeder, W., Martin, K., and Lorensen, B. 1996. *The Visualization Toolkit: An Object Oriented Approach to 3-D Graphics*. Prentice Hall: Englewood Cliffs, NJ.

Seibert, M. and Waxman, A.M. 1992. Adaptive 3-D object recognition from multiple views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:107–124.

Sethian, J.A. 1996. *Level Set Methods*. Cambridge University Press: Cambridge, UK.

Sinha, S.S. and Jain, R. 1994. Range image analysis. In *Handbook of Pattern Recognition and Image Processing: Computer Vision*, vol. 2, T.Y. Young (Ed.). pp. 185–237.

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

Soroka, B.I. and Bajcsy, R.K. 1976. Generalized cylinders from serial sections. In *Proc. International Joint Conference on Pattern Recognition*, pp. 734–735.

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

Stoddart, A.J. and Baker, M. 1998. Reconstruction of smooth surfaces with arbitrary topology adaptive splines. In *Proc. European Conference on Computer Vision*, pp. II:241–254.

Suetens, P., Fua, P., and Hanson, A.J. 1992. Computational strategies for object recognition. *ACM Computing Surveys*, 24(2):5–61.

Swets, D.L. 1996. The self-organising hierarchical optimal subspace learning and inference framework for object recognition. Ph.D. Thesis, Dept. of Computer Science, Michigan State University, East Lansing, Michigan.

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

ter Haar Romeny, B.M. 1994. *Geometry Driven Diffusion in Computer Vision*. Kluwer Academic: Norwell, MA.

Yuen, P., Mokhtarian, F., and Khalili, N. 1999. Multi-scale 3-D surface description: Open and closed surfaces. In *Scandinavian Conference on Image Analysis*, Greenland, pp. 303–310.