Computer Vision and Image Understanding **83,** 118–139 (2001)
doi:10.1006/cviu.2001.0919, available online at http://www.idealibrary.com on IDEAL®

Curvature Computation on Free-Form 3-D Meshes at Multiple Scales

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, United Kingdom
E-mail: F.Mokhtarian@ee.surrey.ac.uk, N.Khalili@ee.surrey.ac.uk, P.Yuen@ee.surrey.ac.uk.

Received May 18, 1999; accepted April 4, 2001

A novel technique for multiscale curvature computation on a smoothed 3-D surface is presented. This is achieved by iteratively convolving local parameterizations of the surface with 2-D Gaussian filters. In our technique, semigeodesic coordinates are constructed at each vertex of the mesh which becomes the local origin. A geodesic from the origin is first constructed in an arbitrary direction such as the direction of one of the incident edges. The smoothing eliminates surface noise and small surface detail gradually and results in gradual simplification of the object shape. The surface Gaussian and mean curvature values are estimated accurately at multiple scales together with curvature zero-crossing contours. The curvature values are then mapped to colors and displayed directly on the surface. Furthermore, maxima of Gaussian and mean curvatures are also located and displayed on the surface. These features have been utilized by later processes for robust surface matching and object recognition. 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. Another advantage is that it is applicable to incomplete surfaces which arise during occlusion or to surfaces with holes.

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 [18] provides several measures of curvature, which include Gaussian and mean curvatures. The combination of these curvature values enables the local surface type to be categorized.

This paper introduces a new technique for multiscale curvature computation on a smoothed 3-D surface. Complete triangulated models of 3-D objects are constructed and using a local parameterization technique are then smoothed using a 2-D Gaussian filter.

1077-3142/01 $35.00

The technique considered here is a generalization of earlier multiscale representation theories proposed for 2-D contours [38] and space curves [33]. More details of the diffusion technique as well as literature survey appear in [34].

In our approach, diffusion of the surface is achieved through convolutions of local parameterizations of the surface with a 2-D Gaussian filter [34, 35, 59]. *Semigeodesic coordinates* [18] are utilized as a natural and efficient way of locally parameterizing surface shape. The most important advantage of our method is that unlike other diffusion techniques such as volumetric diffusion [26, 28] or level set methods [46], 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 the presence of occlusions. It is also more efficient than those techniques since 2-D rather than 3-D convolutions are employed.

The paper is concluded with examples showing simple and complex 3-D objects with their Gaussian and mean curvature values estimated [18]. To visualize these curvature values on the surface, they are then mapped to colors. Color mapping is a scalar visualization technique provided in a software package called Visualisation Toolkit (VTK) [44]. Once surface curvatures are estimated, then curvature zero-crossing contours are recovered and displayed on the surface. Finally, maxima of Gaussian and mean curvatures are also located and displayed on the surface.

The organization 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 covers the computation of Gaussian and mean curvatures as well as their zero-crossing contours and maxima. Section 5 contains a summary of a 3-D object recognition system. Section 6 presents the results and discussion. Section 7 contains the concluding remarks.

2. LITERATURE SURVEY

As the most important application of our work is 3-D surface representation and object recognition, this section presents a survey of previous work on those areas. Many object recognition systems rely on restrictions imposed on the geometry of the object. However, complex free-form surfaces may not be modeled 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 [2]. 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 the inspection of arbitrary curved surfaces and path planning for robot navigation.

Sinha and Jain [47] provide an overview of geometry-based representations derived from the range data of objects. Comprehensive surveys of 3-D object recognition systems are presented by Besl and Jain [3] and Chin and Dyer [8]. Some representation schemes for 3-D objects have adopted some form of surface or volumetric parametric models to characterize 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 [40, 48, 6, 43]. 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 [14, 17] 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 [20, 21] and the four sided spline patches. Triangular meshes have been utilized in our work.

Polyhedral approximations [15] 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 [51] 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. *Generalized cones* or *cylinders* as well as *geons* [41] approximate a 3-D object using globally parameterized mathematical models, but they are not applicable to detailed free-form objects. *Multiview* representations [45] are based on a large number of views of a 3-D object obtained from different viewpoints, but difficulties can arise when a nonstandard view is encountered. In *volumetric diffusion* [26] or level set methods [46], 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 the presence of occlusion. A form of 3-D surface smoothing has been carried out in [54, 55] but this method has drawbacks since it is based on weighted averaging using neighboring vertices and is therefore dependent on the underlying triangulation. Problems can arise when the same surface is triangulated in different ways (for example, as a result of the application of decimation algorithms) resulting in different meshes. As a result, the outcome for each mesh will be different, although they all correspond to the same surface. The smoothing of 3-D surfaces is a result of the diffusion process [56]. For parameterization of a 3-D surface other methods have also been studied, such as the asymptotic coordinates [29], isothermic coordinates [7, 18], and global coordinates [5] used for closed, simply connected objects.

Global representations such as the extended Gaussian image (EGI) [24, 30] 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-centered representations. The graph approach [27] attempts to group a set of infinite 2-D views of a 3-D object into a set of meaningful clusters of appearances. Murase and Nayar [39] and Swets [53] also exploit photometric information to describe and recognize objects. A major drawback of view-centered representations is a lack of complete information. Part-based representations capture structure in object descriptions [11], but there is a lack of agreement in deciding the general set of part primitives to be used which would be sufficient and appropriate. Furthermore, the computation of parts from a single view of an object is difficult.

Recent approaches using splash and super-polygonal segments [50] and algebraic polynomials [42] 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. Note that a substantial amount of work has also been carried out on curvature estimation from range images [4, 16, 31, 52, 57]. However, a range image corresponds to just one view of a 3-D object. It is essentially a 2-D function with known

global parameterization, whereas a 3-D mesh is substantially more complex. Finally, mesh simplification techniques [9, 10, 19, 22, 49] are not suitable for object recognition applications since the locations of mesh vertices chosen at higher scales tend to be arbitrary and not robust to noise or local variations of shape.

3. SEMIGEODESIC PARAMETERIZATION

Free-form 3-D surfaces are complex; hence, no global coordinate system exists on these surfaces which could yield a natural parameterization 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* [18]. 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 [34].

3.1. Geodesic Lines and Semigeodesic Coordinates

A geodesic line is defined as a contour which locally represents the shortest distance on a 3-D surface between any two points on that contour. Initially a geodesic line is drawn arbitrarily through the origin of the local area. This geodesic line is sampled at equal-sized intervals based on the average length of all triangle edges on the mesh. No interpolation is carried out to compute the 3-D coordinates of sampled points. The second family of lines are also geodesic lines. All the lines together form semigeodesic coordinates. Note that semigeodesic coordinates are constructed at every vertex on the mesh.

3.2. Geodesic Line Construction

Before semigeodesic coordinates can be generated on a local patch at a chosen vertex V, an arbitrary geodesic line is required. The edge connecting V and one of its neighboring vertices is selected as the arbitrary direction. Once this direction is determined, the next step is to construct a geodesic line. This line is constructed on the local 3-D surface by following the geodesic path in a straight line until an edge or vertex is reached. This new edge point or vertex becomes a starting point for the next extension. To continue a geodesic line into the next triangle, we first measure the angle between the path and the common edge which the path has intersected. We then extend the path to the next triangle using the same angle. The construction of this geodesic line continues until the last edge or vertex of the local area is reached. Same process is repeated to construct the reverse direction of the geodesic line.

3.3. Perpendicular Geodesic Line Construction

The second family of lines is constructed perpendicularly to the above newly created geodesic line. Each sampled point of that geodesic line is used as a reference point for constructing these perpendicular geodesic lines. Using a similar technique as described in Section 3.2, the perpendicular lines are constructed in the forward direction as well as in

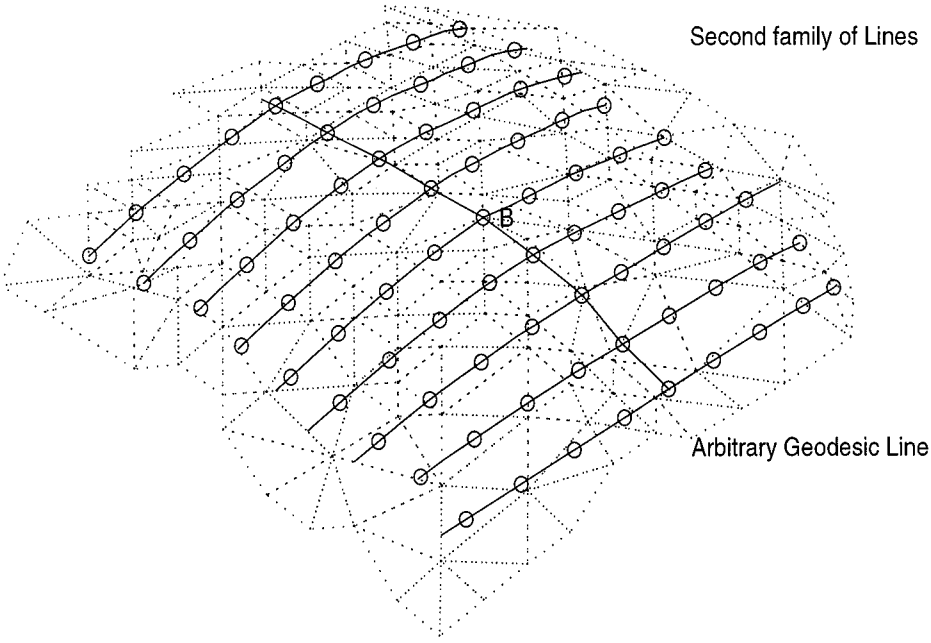


FIG. 1. Semigeodesic coordinates on a triangular mesh, where o's are the semigeodesic coordinates and B is the current vertex.

the backward direction with respect to the geodesic line. This completes the construction of local semigeodesic parameterization. Semigeodesic coordinates can also be constructed at or near a boundary in case of an incomplete surface or a surface with holes. In such cases, geodesic lines are constructed as before but they are terminated as soon as they intersect the surface boundary. Figure 1 shows the complete semigeodesic coordinates on a triangular mesh.

3.4. 2-D Gaussian Convolution

Gaussian filtering is a weighted average smoothing carried out at a vertex and its neighborhood. A 2-D Gaussian filter is generated according to the formula below [1]:

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

The result of smoothing depends entirely on a vertex and its neighborhood. So the filtering uses a *local area* of the surface with its size the same as the filter size. A large 3-D triangulated surface is overlaid by many fixed size small local areas. In order to eliminate over- and undersampling, the area size must be neither too large nor too small. In other words, a local area must cover a reasonable size neighborhood in order to provide accurate results. Experiments were conducted with a filter size of 9 (optimal filter size). In other words, the Gaussian was approximated by a square mask with nine elements in each direction. The value of σ was set to 1.0 in the formula for the Gaussian filter. The unit used for sampling the mesh was equal to the average value of the lengths of all triangle edges on mesh. The meshes are normalized to avoid scale problems.

In order to smooth a 3-D surface, a fixed size 2-D Gaussian filter is convolved with the local area. Local parameterization of the surface yields:

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

The smooth surface is defined by

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

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. CURVATURE ESTIMATION

This section presents techniques for accurate estimation of Gaussian and mean curvatures at multiple scales on smoothed free-form 3-D surfaces. Differential geometry provides several measures of curvature, which include Gaussian and mean curvatures [18]. 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 regualar 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:

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

Both Gaussian and mean curvature values are direction-free quantities. Gaussian and mean curvatures are invariant to arbitrary transformations of the (u, v) parameters as well as rotations and translations of a surface. A combination of these curvature measures enables the local surface type to be categorized. On smoothed surfaces of 3-D objects, the procedure for estimating the Gaussian and mean curvatures is as follows. For each point of the surface,

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

the corresponding local neighborhood data are convolved with the partial derivatives of the Gaussian function $G(u, v, \sigma)$; *i.e.,*

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

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

4.1. Curvature Zero-Crossing Contours

Having computed curvature values at each vertex of a smoothed 3-D surface, one can locate curvature zero-crossing contours where curvature functions K or H defined by Eqs. (4) and (5) are equal to zero. Curvature zero-crossing contours can be useful for segmenting a smoothed 3-D surface into regions. The process of recovery of the curvature zero-crossing contours is identical for Gaussian and mean curvatures. Every edge e of the smoothed surface is examined in turn. If the vertices of e have the same signs of curvature, then there is no curvature zero-crossing point on e. However, if the vertices of e have different signs of curvature, then there exists a point on e at which curvature goes to zero. The zero-crossing point is assumed to be at the midpoint of e. The other two edges of the triangle to which e belongs will then be checked since there will be another zero-crossing point

on one of those edges. When that zero-crossing is found, it is connected to the previously found zero-crossing. The curvature zero-crossing contour is tracked in this fashion until one arrives back at the starting point.

4.2. 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 neighbors of V are defined as vertices which are connected to V by an edge. If the curvature value of V is higher than the curvature values of all its neighbors, V is marked as a local maximum of curvature. Curvature maxima can be utilized by later processes for robust surface matching and object recognition with occlusion.

5. RELIABLE FREE-FORM 3-D OBJECT RECOGNITION

A system for free-form 3-D object recognition using 3-D models has been developed [36, 37]. The system is based on the technique presented in this paper. Smoothing is utilized to remove noise and to reduce the number of feature points to add to the efficiency and robustness of the system. In order to find the optimal smoothing scale for each object, we employed the following multiscale 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 minimized. This indicates that the features have become stable at the corresponding scale which is used as the optimal scale for smoothing.

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. It should be noted that object disconnection does not affect our recognition system. The reason is that feature detection is quite local and therefore not influenced by any disconnection of the objects. As a result, matching can take place even with disconnected objects.

Recognition results indicated that the system performed robustly and efficiently. The matching system was implemented entirely in C++ and run on an UltraSparc 170E. The system was quite fast with matching times not exceeding 2–3 CPU s in each case. Three different experiments were devised to evaluate the system. The first experiment consisted of applying arbitrary amounts of scaling and 3-D rotation to the database objects. All objects were subsequently recognized correctly by the system. The second experiment made use of incomplete surfaces which were again subjected to arbitrary amounts of scaling and 3-D rotation. In order to obtain incomplete surfaces, up to 60% of connected vertices were removed from database objects. As in the previous experiment, all input objects were correctly recognized by the system. In the third experiment, three complex scenes were created, each consisting of two or more objects. The system was able to recognize the major objects in each of those scenes. However, very small objects or objects with many missing features posed difficulties for the recognition system.

6. RESULTS AND DISCUSSION

This section presents some results on free-form surface smoothing as well as curvature estimation.

6.1. Diffusion

The smoothing routines were implemented entirely in C++. Complete triangulated models of 3-D objects used for our experiments are constructed at our center [21]. In order

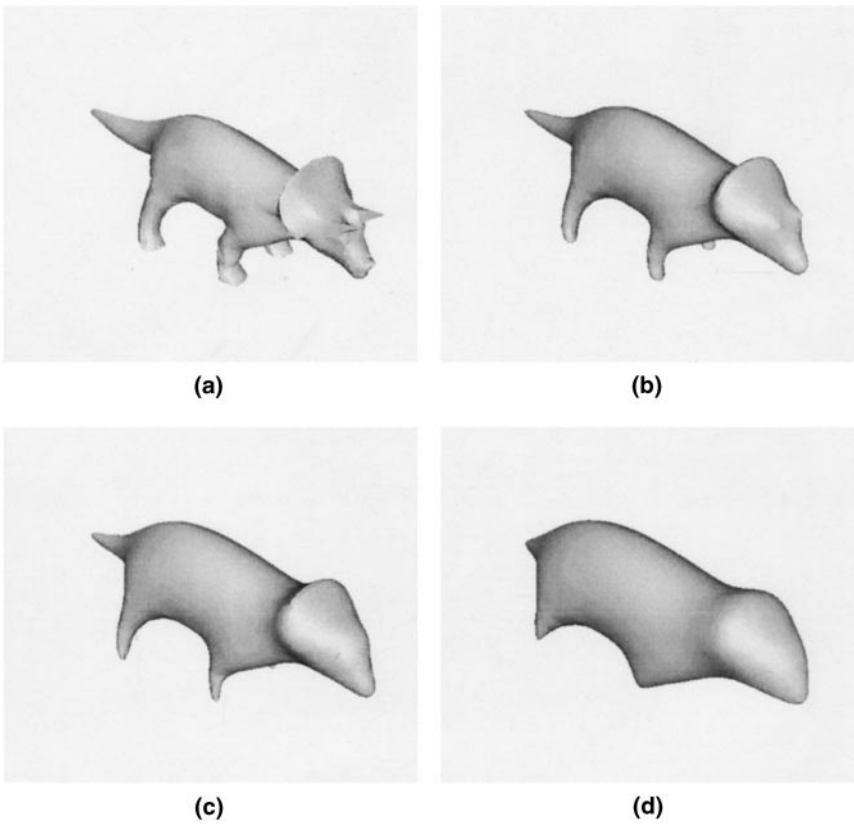


FIG. 2. Smoothing of the dinosaur. (a) Original, (b) three iterations, (c) five iterations, (d) ten iterations.

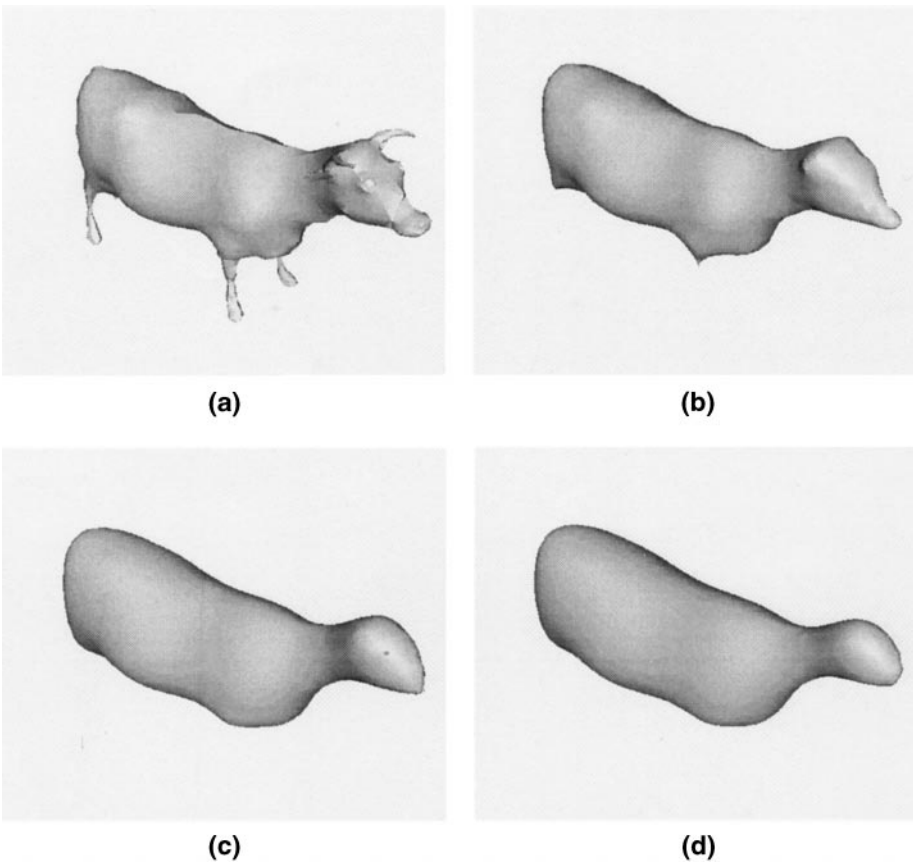


FIG. 3. Smoothing of the cow. (a) Original, (b) three iterations, (c) eight iterations, (d) twelve iterations.

to experiment with our techniques, both simple and complex 3-D objects with different numbers of triangles were used. Each iteration of smoothing of a surface with 1000 vertices takes about 0.5 s of CPU time on an UltraSparc 170E.

The first test object was a dinosaur with 2996 triangles and 1500 vertices as shown in Fig. 2. The object becomes smoother gradually and the legs, tail, and ears are removed after 10 iterations. The next test object was a cow with 3348 triangles and 1676 vertices as shown in Fig. 3. The surface noise is eliminated iteratively with the object becoming smoother gradually where after 12 iterations the legs, ears, and tail are removed, as was seen for the dinosaur.

Our smoothing technique was also applied to a number of open–incomplete surfaces. Figure 4 shows the results obtained on a part of a telephone handset. This object also has a triangle removed in order to generate an internal hole. Figure 5 shows the smoothing results obtained on the lower part of a chair object. Figure 6 shows smoothing results obtained on a partial rabbit. The object is smoothed iteratively and the ears disappear eventually.

These examples show that our technique is effective in substantial elimination of surface noise as well as removal of surface detail. The result is gradual simplification of object shape. 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.

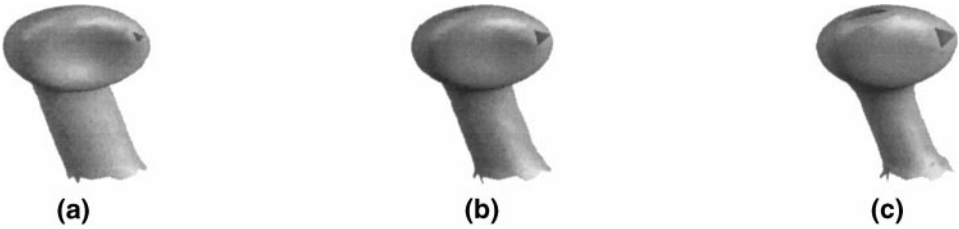


FIG. 4. Diffusion of the partial telephone handset. (a) Original, (b) three iterations, (c) five iterations.

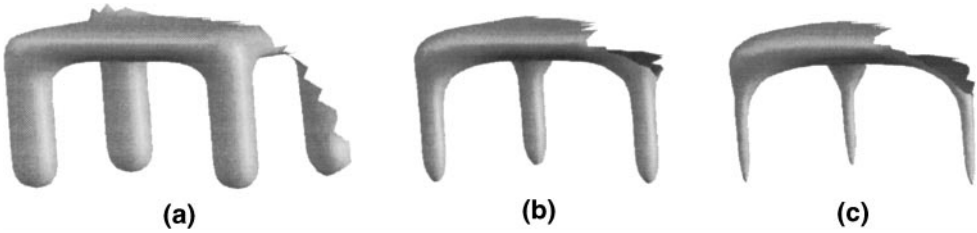


FIG. 5. Smoothing of the partial chair. (a) Original, (b) three iterations, (c) four iterations.

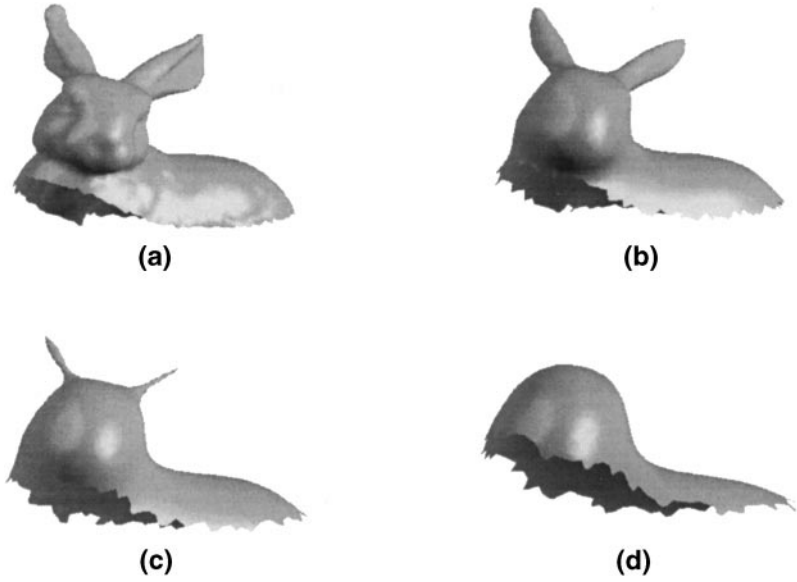
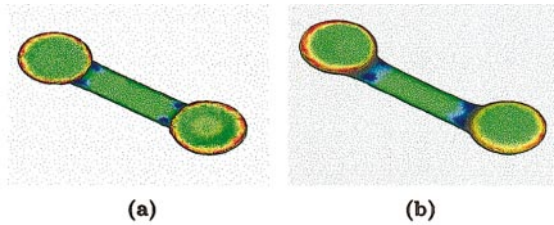
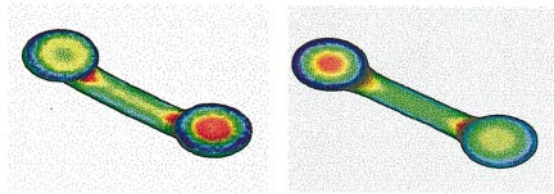


FIG. 6. Smoothing of the rabbit. (a) Original, (b) five iterations, (c) nine iterations, (d) 15 iterations.

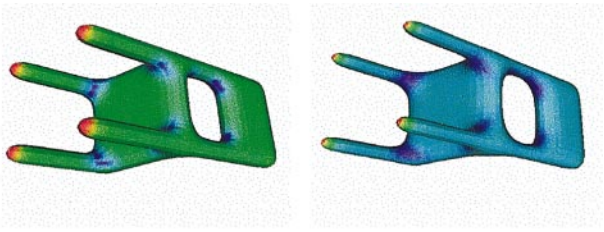


(a) (b)

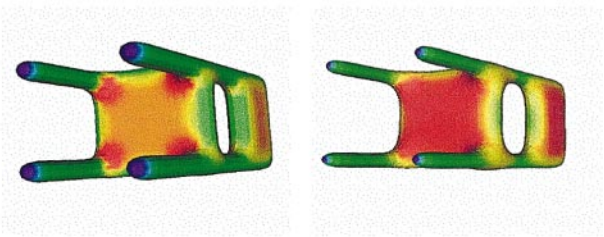


(c) (d)

7

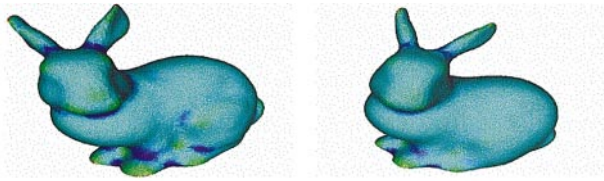


(a) (b)

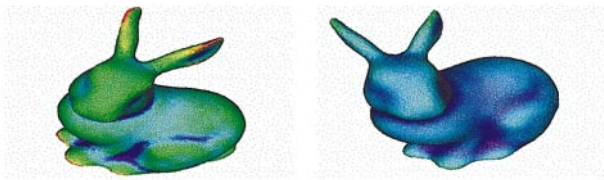


(c) (d)

8

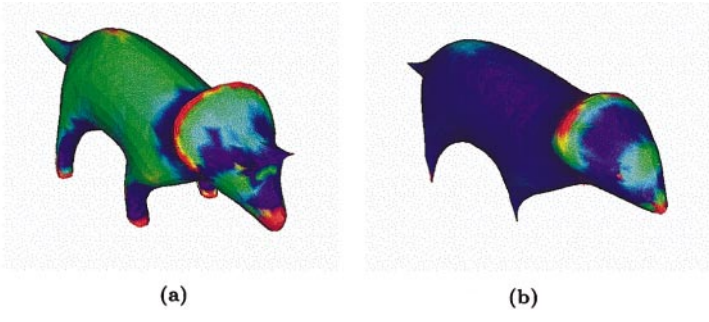


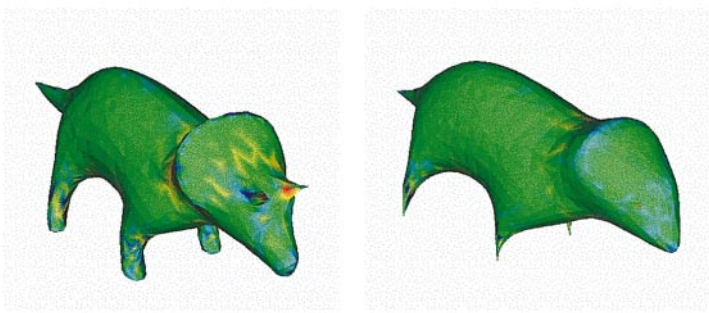
(a) (b)



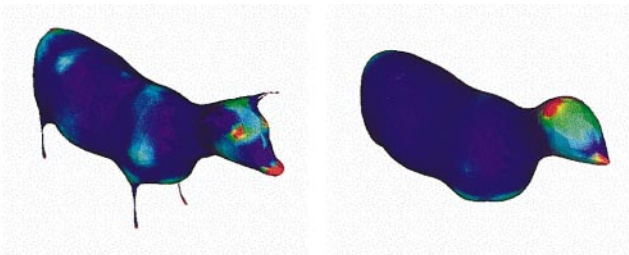
(c) (d)

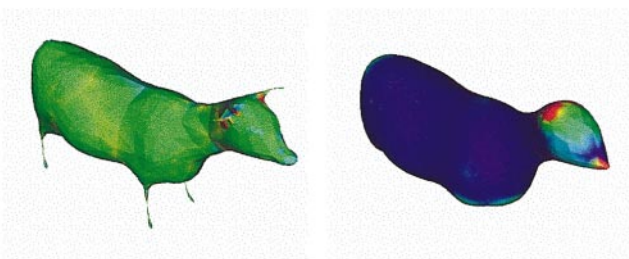
9





10





11

FIG. 7. Gaussian (top row) and mean (bottom row) curvatures on the phone handset. (a, c) One iteration, (b, d) two iterations.

FIG. 8. Gaussian (top row) and mean (bottom row) curvatures on the chair. (a, c) One iteration, (b, d) two iterations.

FIG. 9. Gaussian (top row) and mean (bottom row) curvatures on the rabbit. (a, c) One iteration, (b, d) two iterations.

FIG. 10. Gaussian (top row) and mean (bottom row) curvatures on the dinosaur. (a, c) One iteration, (b, d) six iterations.

FIG. 11. Gaussian (top row) and mean (bottom row) curvatures on the cow. (a, c) One iteration, (b, d) six iterations.

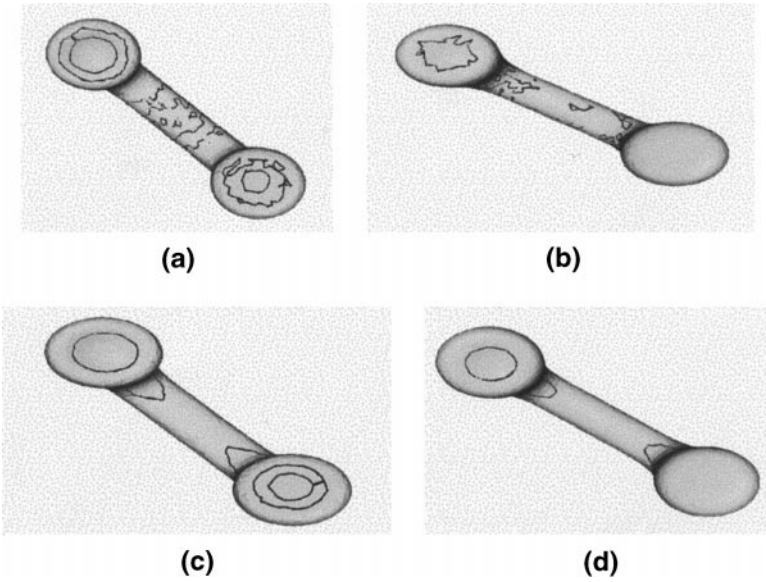


FIG. 12. Gaussian (top row) and mean (bottom row) curvature zero-crossing contours on the phone handset. (a, c) One iteration, (b, d) three iterations.

Our technique overcomes this problem by integrating surface smoothing and curvature estimation into one unified formalism. This technique was subjected to further testing using noisy meshes. The noise has been introduced by the range-finder and by the process of range image fusion to build 3D models. The results can be observed in [34] and [59].

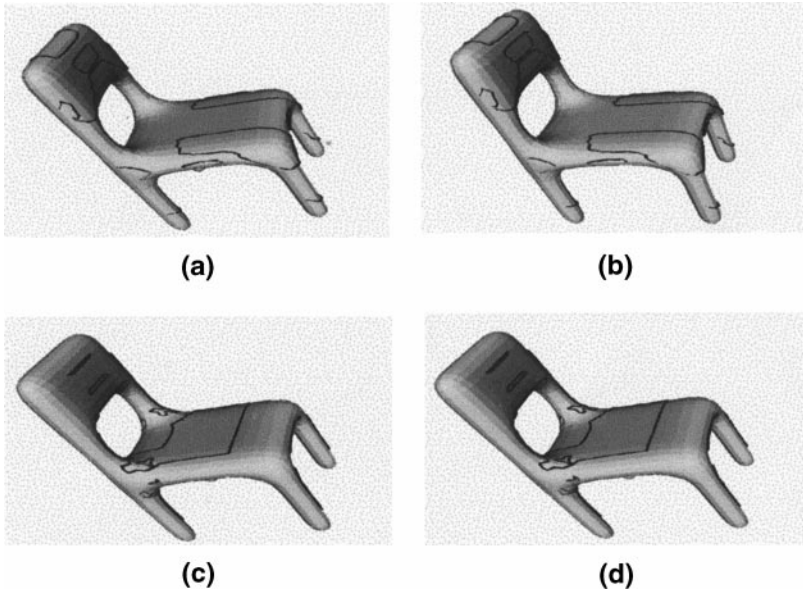


FIG. 13. Gaussian (top row) and mean (bottom row) curvature zero-crossing contours on the chair. (a, c) One iteration, (b, d) two iterations.

This technique is not volume preserving in its current form. We do not consider this to be important. However, if it is important in a given application, the object size can be adjusted after each iteration of smoothing to ensure that its volume is preserved.

It is possible to produce disconnected components of a particular initial object. This is a natural outcome of the smoothing process and occurs at points where the object happens to be thin. A decimation algorithm [23] is applied after each iteration of smoothing to remove odd triangles. The same process can segment the object at points where it becomes very thin [25]. We believe that the disconnection of data is desirable when objects become very thin at specific points on their surfaces. This phenomenon can be considered a natural consequence of the multi-scale description of a 3-D object: at larger scales, the smaller or thinner parts of the object tend to disappear, and only the major parts survive. Therefore, this outcome should not be avoided.

6.2. Curvature Estimation

This section presents the results of application of our curvature estimation techniques to 3-D objects using methods described in Section 4. The diffusion results for other 3-D surfaces were given in [34]. The first example is a phone handset. After smoothing the object, the Gaussian curvatures of all vertices were estimated. To visualize these curvature values on the surface, they are then mapped to colors using the VTK [44], and the results are shown in Fig. 7a. Surface curvature colors are coded as follows: red, high; blue, low; and other colors designate nonextreme values. All convex corners of the phone handset 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 phone handset and the results are shown in Fig. 7b. This indicates that mean curvature values for the edges are different than those for flat areas, as expected.

The next object was a chair. Its Gaussian curvature values were estimated and results are shown in Fig. 8a. These results again confirm that the curvature values are high and low at convex and concave corners, respectively, and close to zero on the flat regions such as the seat and the back. The mean curvatures of the chair were also estimated and the results are shown in Fig. 8b. For more complex objects the Gaussian and mean curvatures were also estimated and Figs. 9, 10, and 11 show the results for a rabbit, a dinosaur, and a cow, respectively.

Next, the curvature zero-crossing contours of these surfaces were found and displayed on the surface using VTK. Curvature zero-crossing contours can be used for segmenting surfaces into regions. Figure 12a shows Gaussian curvature zero-crossing contours for the smoothed phone handset. Figure 12b shows mean curvature zero-crossing contours for the same object.

Figure 13a shows the Gaussian curvature zero-crossing contours for the smoothed chair and Fig. 13b shows the mean curvature zero-crossing contours for the same object. Figure 14 shows Gaussian and mean curvature zero-crossing contours for the smoothed rabbit. The same experiments were also repeated for the dinosaur and cow, and these results are shown in Figs. 15 and 16, respectively. Notice that the number of curvature zero-crossing contours are reduced as the object is smoothed iteratively.

Finally, the local curvature maxima for the smoothed telephone handset are computed. The local maxima of Gaussian curvature are displayed on the surface as shown in Fig. 17a. Figure 17b shows the local maxima of mean curvature for the same object. Figure 18a

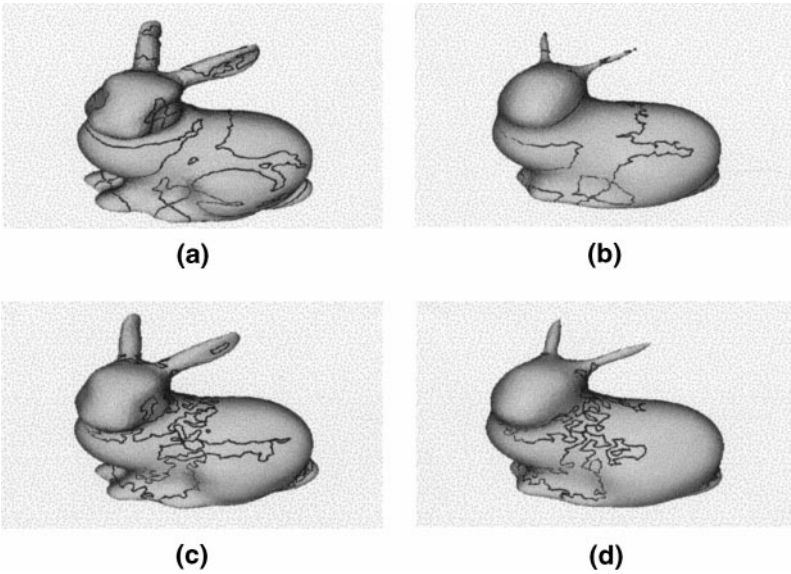


FIG. 14. Gaussian (top row) and mean (bottom row) curvature zero-crossing contours on the rabbit. (a, c) One iteration, (b, d) six iterations.

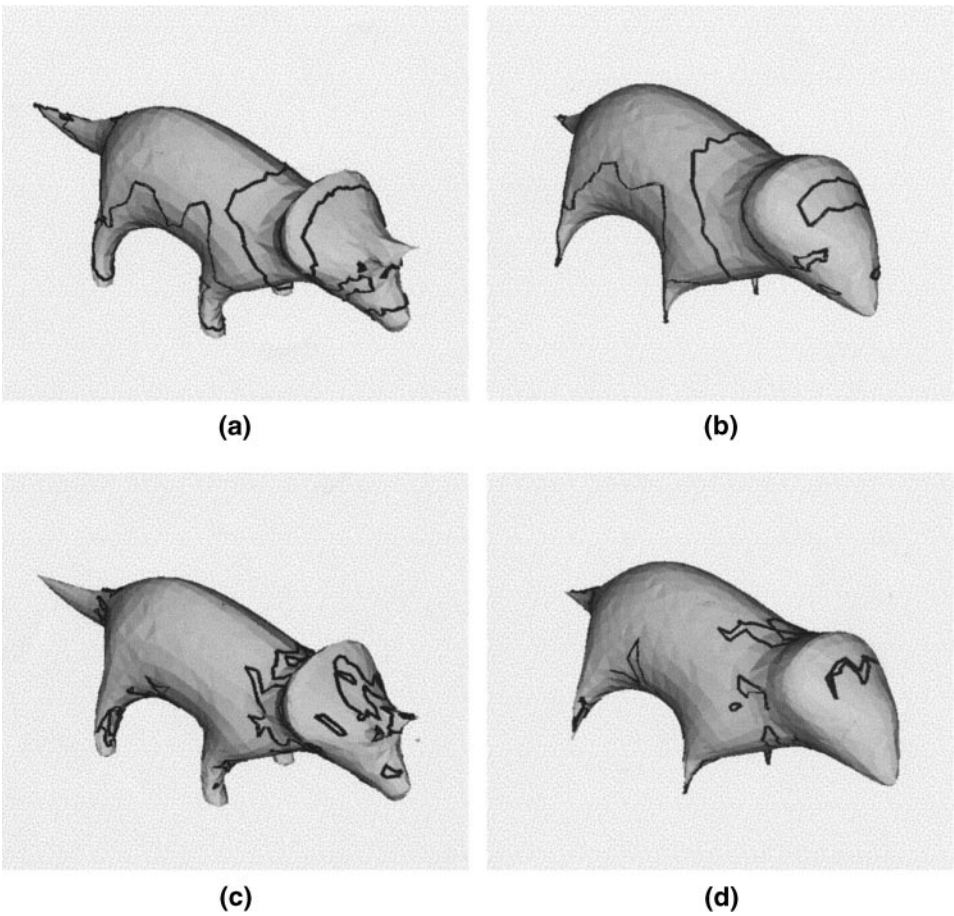


FIG. 15. Gaussian (top row) and mean (bottom row) curvature zero-crossing contours on the dinosaur. (a, c) One iteration, (b, d) six iterations.

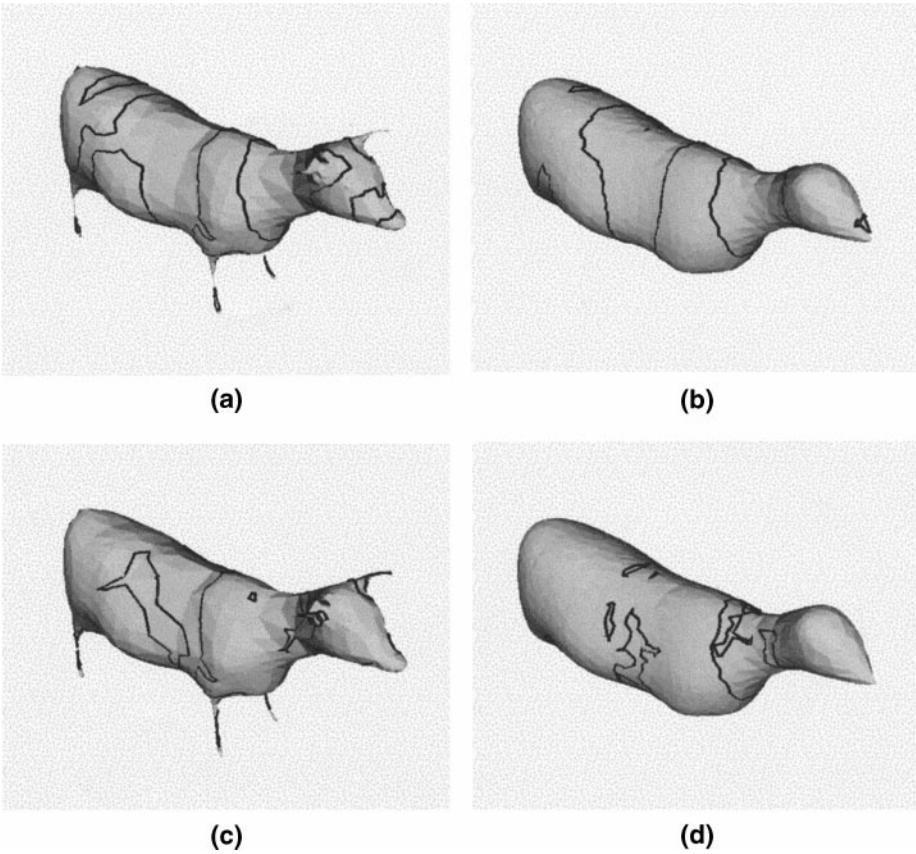


FIG. 16. Gaussian (top row) and mean (bottom row) curvature zero-crossing contours on the cow. (a, c) One iteration, (b, d) six iterations.

shows the local maxima of Gaussian curvature for the smoothed chair, and Fig. 18b shows the local maxima of mean curvature for the same object. Note that some of the curvature maxima are not visible due to self-occlusion.

Figures 19a and 19b show the local maxima of Gaussian and mean curvatures for the rabbit. The local maxima of Gaussian and mean curvatures for the dinosaur and cow are also shown in Figs. 20 and 21, respectively. All curvature maxima are shown after one

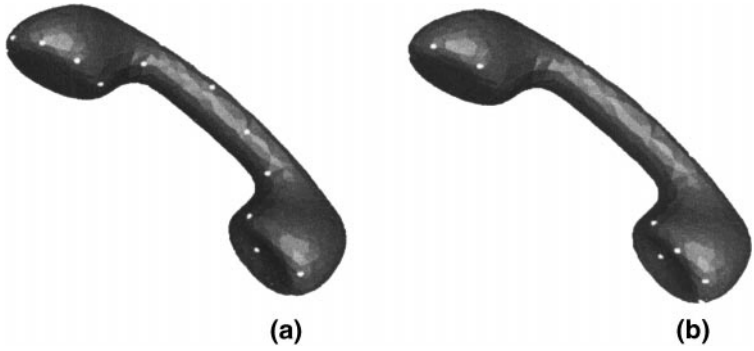


FIG. 17. Curvature maxima of the phone handset. (a) Gaussian, (b) mean.

FIG. 18. Curvature maxima of the Chair. (a) Gaussian, (b) mean.

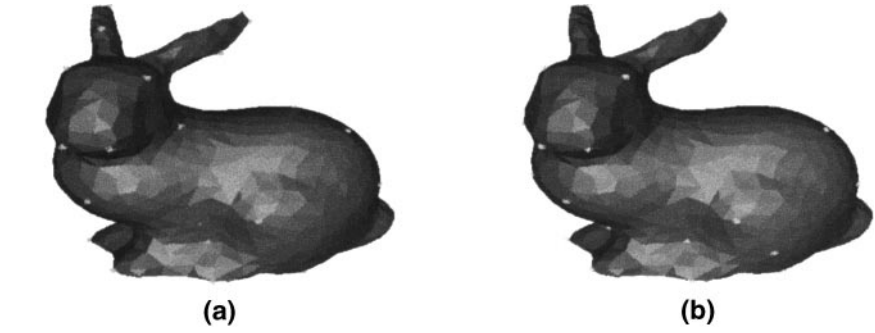


FIG. 19. Curvature maxima of the rabbit. (a) Gaussian, (b) mean.

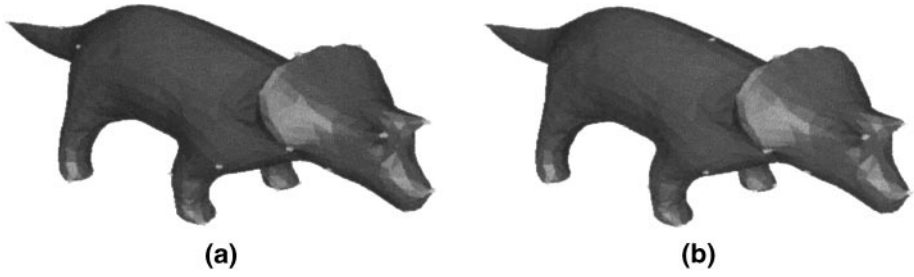


FIG. 20. Curvature maxima of the dinosaur. (a) Gaussian, (b) mean.

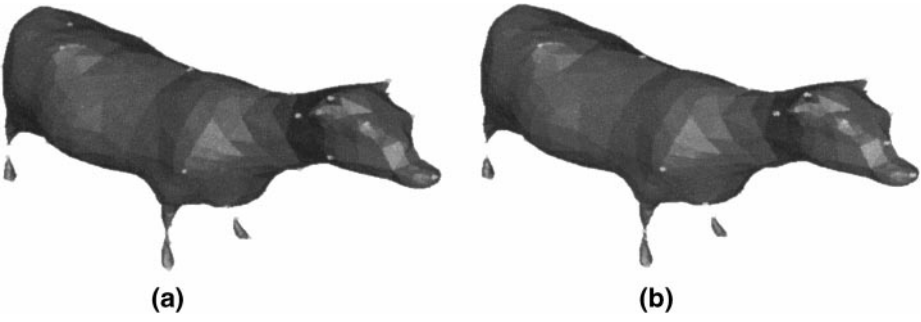


FIG. 21. Curvature maxima of the cow. (a) Gaussian, (b) mean.

TABLE 1
Comparison of Other Techniques

Technique	Short analysis
Range image analysis	Applies only to a single-view range image but not a free-form 3-D mesh structure.
Mesh neighborhood weighted averaging	(1) Outcome depends on how the surface is triangulated. (2) Does not provide a method of curvature estimation.
Level set methods and volumetric diffusion	(1) They lack local support so not suitable for incomplete surfaces–object recognition with occlusion. (2) Inefficient since problem dimensionality is increased.
Mesh simplification	(1) Mesh vertices not robust to noise–shape distortions. (2) Does not provide a way for curvature estimation.
Proposed method	(1) Combines multiscale description and curvature estimation. (2) Applicable to noisy, free-form, incomplete meshes. (3) Efficient and robust to mesh geometry.

iteration. These features can be utilized by later processes for robust surface matching and object recognition with occlusion. As these examples show, we believe that multiscale feature recovery is more robust feature extraction at a single scale [32]. Animation of surface diffusion can be observed at the Web site: http://www.ee.surrey.ac.uk/Research/VSSP/demos/css3d/index.html.

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. However, we believe that the use of simple objects with known curvatures is not a very satisfactory method of testing our method since our technique was designed for curvature estimation on noisy, free-form meshes with unknown curvature values. It should be noted that the best evidence in favor of the accuracy of our method is the comprehensive work by Estrozi *et al.* on 1-D and 2-D approaches to numeric curvature estimation and their comparative performance assessment [12, 13].

We also devised another experiment which showed our method can estimate curvature values accurately regardless of how semigeodesic coordinates are constructed on the surface. Due to space constraints, that experiment was not described in this paper but a full description can be found in [58].

Table 1 contains a comparision of our multiscale technique for free-form 3-D surface description with other relevant methods.

7. CONCLUSIONS

A novel technique for multiscale curvature computation on a smoothed 3-D surface was presented. In our technique semigeodesic coordinates were constructed at each vertex of the mesh which becomes 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 parameterization 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. The surface Gaussian and mean curvatures were also estimated. To visualize these curvature values on the surface, they were then mapped to colors and shown directly on the surface provided in the Visualisation Toolkit. 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.

Next, the Gaussian and mean curvature zero-crossing contours were also recovered and displayed on the surface. Results indicated that as the surface is smoothed iteratively, the number of curvature zero-crossing contours were reduced. Curvature zero-crossing contours can be used for segmenting surfaces into regions. Furthermore, the local maxima of Gaussian and mean curvatures were also located and displayed on the surface. These features can be utilized by later processes for robust surface matching and object recognition with occlusion.

REFERENCES

1. S. P. Banks, *Signal Processing, Image Processing and Pattern Recognition*, Prentice-Hall, Englewood Cliffs, N.J., 1990.
2. P. J. Besl, The free-form surface matching problem, in *Machine Vision for Three-dimensional Scenes* (H. Freeman, Ed.), pp. 25–71, 1990.
3. P. J. Besl and R. C. Jain, Three dimentional object recognition, *ACM Comput. Surveys* **17**, 1985, 75-145.
4. M. Brady, J. Ponce, A. Yuille, and H. Asada, Describing surfaces, *Comput. Vision Graphics Image Process.* **32**, 1985, 1–28.
5. C. Brechbuhler, G. Gerig, and O. Kubler, Parametrization of closed surfaces for 3-d shape description, *Comput. Vision Image Understanding* **61**, 1995, 154–170.
6. T. W. Chen and W. C. Lin, A. neural network approach to CSG-based 3-D object recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 719–726.
7. S. S. Chern, P. Hartman, and A. Wintner, On isometric coordinates, in *Commentaries Mathematical Helvetici*, Vol. 28, 1954.
8. R. T. Chin and C. R. Dyer, Model-based recognition in robot vision, *ACM Comput. Surveys* **18**, 1986, 67–108.
9. K. Choo, I. D. Yun, and S. U. Lee, Edge-based approach to mesh simplification, in *International Conference on 3-D Digital Imaging and Modeling, 1999.*
10. J. Cohen, Simplification envelopes, in *Proc SIGGRAPH, 1996.*
11. S. J. Dickinson, A. P. Pentland, and A. Rosenfeld, 3-d shape recovery using distributed aspect matching, *IEEE, Trans. Pattern Anal. Mach. Intell.* **14**, 1992, 174–198.
12. L. F. Estrozi, A. G. Campos, L. G. Rios, R. M. Cesar Jr., and L. da F. Costa, Comparing curvature estimation techniques, in *Proc 4th Simpósio Brasileiro de Automação Inteligente, São Paulo, Brazil, 1999*, pp. 58–63.
13. L. F. Estrozi, L. G. Rios, A. G. Campos, R. M. Cesar Jr., and L. da F. Costa, 1D and 2D Fourier-based approaches to numeric curvature estimation and their comparative performance assessment. *Graphical Models*, 2001, Submitted for publication.
14. T. J. Fan, G. Medioni, and R. Nevatia, Recognising 3-d objects using surface descriptions, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 1140–1157.
15. O. D. Faugeras and M. Hebert, The representation, recognition, and locating of 3-D objects, *Internat. J. Robotics Res.* **5**(3) 1986, 27–52.
16. P. J. Flynn and A. K. Jain, On reliable curvature estimation, in *Proc IEEE Conference on Computer Vision and Pattern Recognition*, pp. 110–116, 1989.
17. P. J. Flynn and A. K. Jain, Bonsai: 3-d object recognition using constrained search, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 1991, 1066–1075.
18. A. Goetz, *Introduction to Differential Geometry*, Addison-Wesley, Reading, MA, 1970.
19. I. Guskov, W. Sweldens, and P. Schroder, Multiresolution signal processing for meshes, in *Proc SIGGRAPH*, 1999.

20. A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, Marching triangles: Range image fusion for complex object modelling, in *Proc IEEE International Conference on Image Processing, Lausanne, Switzerland, 1996*, pp. 381–384.
21. A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, Reliable surface reconstruction from multiple range images, in *Proc European Conference on Computer Vision, Cambridge, UK, 1996.* pp. 117–126.
22. H. Hoppe, Mesh optimization, in *Proc SIGGRAPH, 1993.*
23. H. Hoppe, Progressive meshes, in *Proc SIGGRAPH*, pp. 99–106, 1996.
24. S. B. Kang and K. Ikeuchi, The complex egi, a new representation for 3-d pose determination, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 1993, 707–721.
25. N. Khalili, F. Mokhtarian, and P. Yuen, Free-form surface description in multiple scales: Extension to incomplete surfaces, in *Proc. International Conference on Computer Analysis of Images and Patterns, Slovenia, Ljubljana, 1999*, pp. 293–300.
26. J. J. Koenderink, *Solid Shape*, MIT Press, Cambridge, MA, 1990.
27. J. J. Koenderink and A. J. vanDoorn, Internal representation of solid shape with respect to vision, *Biol. Cybernet.* **32**, 1979, 211–216.
28. J. J. Koenderink and A. J. vanDoorn, Dynamic shape, *Biol. Cybernet.* **53**, 1986, 383–396.
29. E. Kreyszig, *Differential Geometry*, Oxford University Press, Oxford, 1959.
30. P. Liang and C. H. Taubes, Orientation-based differential geometric representations for computer vision applications, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 249–258.
31. P. Liang and J. S. Todhunter, Representation and recognition of surface shapes in range images: A differential geometry approach, *Comput. Vision Graphics Image Process.* **52**, 1990, 78–109.
32. F. Mokhtarian, Silhouette-based object recognition with occlusion through curvature scale space, in *Proc European Conference on Computer Vision, Cambridge, England, 1996*, pp. 566–578.
33. F. Mokhtarian, A theory of multi-scale, torsion-based shape representation for space curves, *Comput. Vision Image Understanding* **68**, 1997, 1–17.
34. F. Mokhtarian, N. Khalili, and P. Yuen, Multi-scale 3-D free-form surface smoothing, in *Proc. British Machine Vision Conference*, pp. 730–739, 1998.
35. F. Mokhtarian, N. Khalili, and P. Yuen, Multi-scale free-form surface description, in *Proc. Indian Conference on Computer Vision, Graphics and Image Processing, New Delhi, India, 1998*, pp. 70–75.
36. F. Mokhtarian, N. Khalili, and P. Yuen, Free-form 3-d object recognition at multiple scales, in *Proc. British Machine Vision Conference, Bristol, 2000*, pp. 446–455.
37. F. Mokhtarian, N. Khalili, and P. Yuen, Multi-scale free-form 3D object recognition using 3D models, *Image Vision Comput.* **19**, 2001, 273–283.
38. F. Mokhtarian and A. K. Mackworth, A theory of multi-scale, curvature-based shape representation for planar curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1992, 789–805.
39. H. Murase and S. K. Nayar, Visual learning and recognition of 3-D objects from appearance, *Internat. J. Comput. Vision* **14**, 1995, 5–24.
40. A. P. Pentland, Perceptual organisation and the representation of natural form, *Artificial Intel.* **28**, 1986, 293–331.
41. M. Pilu and R. Fisher, Recognition of geons by parametric deformable contour models, in *Proc. European Conference on Computer Vision, Cambridge, UK, 1996*, pp. 71–82.
42. J. Ponce, D. J. Kriegman, S. Petitjaan, S. Sullivan, G. Taubin, and B. Vijayakumar, Representations and algorithms for 3-D curved object recognition, in *In Three-dimensional Object Recognition Systems, Amsterdam, Netherlands, 1993*, pp. 17–56.
43. H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
44. W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object Oriented Approach to 3-D Graphics*, Prentice Hall, Englewood Cliffs, NJ, 1996.
45. M. Seibert and A. M. Waxman, Adaptive 3-D object recognition from multiple views, in *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1992, 107–124.
46. J. A. Sethian, *Level Set Methods*, Cambridge University Press, MA, 1996.

47. S. S. Sinha and R. Jain, Range image analysis, in *Handbook of Pattern Recognition and Image Processing: Computer Vision* (T. Y. Young, Ed.), Vol. 2, pp. 185–237, 1994.
48. F. Solina and R. Bajcsy, Recovery of parametric models from range images: The case for superquadrics with global deformations, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 1990, 131–147.
49. M. Soucy and D. Laurendeau, Multiresolution surface modeling based on hierarchical triangulation, *Comput. Vision Image Understanding* **63**, 1996, 1–14.
50. F. Stein and G. Medioni, Structural indexing: Efficient 3-D object recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **4**, 1992, 125–145.
51. A. J. Stoddart and M. Baker, Reconstruction of smooth surfaces with arbitrary topology adaptive splines, in *Proc. European Conference on Computer Vision*, Vol. II, pp. 241–254, 1998.
52. A. J. Stoddart, J. Illingworth, and T. Windeatt, Optimal parameter selection for derivative estimation from range images, *Image Vision Comput.* **13**, 1995, 629–635.
53. D. L. Swets, *The Self-organising Hierarchical Optimal Subspace Learning and Inference Frame-work for Object Recognition*, Ph.D. thesis, Michigan State University, Dept of Computer Science, East Lansing, Michigan, 1996.
54. G. Taubin, Curve and surface smoothing without shrinkage, in *Proc. International Conference on Computer Vision*, pp. 852–857, 1995.
55. G. Taubin, T. Zhang, and G. Golub, Optimal surface smoothing as filter design, in *Proc. European Conference on Computer Vision*, pp. 283–292, 1996.
56. B. M. ter Haar Romeny, *Geometry Driven Diffusion in Computer Vision*, Kluwer Academic, Dordrecht/ Norwell, MA, 1994.
57. B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, Curvature-based representation of objects from range data, *Image Vision Comput.* **4**, 1986, 107–114.
58. P. Yuen, N. Khalili, and F. Mokhtarian, Curvature estimation on smoothed 3-d meshes, in *Proc British Machine Vision Conference*, pp. 133–142, 1999.
59. P. Yuen, F. Mokhtarian, and N. Khalili, Multi-scale 3-d surface description: open and closed surfaces, in *Scandinavian Conference on Image Analysis*, Greenland, 1999, pp. 303–310.