International Journal of Shape Modeling
© World Scientific Publishing Company

ESTIMATING CURVATURE ON TRIANGULAR MESHES

TIMOTHY D. GATZKE
Dept. of Computer Science and Engineering,
Washington University in St. Louis,
One Brookings Drive,
St. Louis, MO 63130, USA
tdg1@cs.wustl.edu

CINDY M. GRIMM
Dept. of Computer Science and Engineering,
Washington University in St. Louis,
One Brookings Drive,
St. Louis, MO 63130, USA
cmg@wustl.edu
http://www.cs.wustl.edu/∼cmg

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

This paper takes a systematic look at methods for estimating the curvature of surfaces represented by triangular meshes. We have developed a suite of test cases for assessing both the detailed behavior of these methods, and the error statistics that occur for samples from a general mesh. Detailed behavior is represented by the sensitivity of curvature calculation methods to noise, mesh resolution, and mesh regularity factors. Statistical analysis breaks out the effects of valence, triangle shape, and curvature sign. These tests are applied to existing discrete curvature approximation techniques and common surface fitting methods. We provide a summary of existing curvature estimation methods, and also look at alternatives to the standard parameterization techniques. The results illustrate the impact of noise and mesh related issues on the accuracy of these methods and provide guidance in choosing an appropriate method for applications requiring curvature estimates.

Keywords: Computational Geometry; Object Modeling; Curve; Surface; Solid.

1991 Mathematics Subject Classification: 22E46, 53C35, 57S20

1. Introduction

There has been substantial growth in the use of polygonal mesh representations for complex free-form shapes. Advances in scanner technology, 3D sensors, etc., and algorithms for constructing meshes from this coordinate data [2,3,21,24], have made models for such objects readily available. Meshes support wide variations in complexity and resolution for local regions of an object. They use a relatively simple representation consisting of vertices (points sampled from the surface), and polygonal faces defining connectivity between vertices. Today's visualization tools are extremely compatible with this mesh data structure. However, tools for extracting surface properties, such as smoothness, from meshes have not yet progressed to match the state-of-the-art for more traditional representations such as those used in the Computer-Aided Design (CAD) environment.

Curvature is an intrinsic property of surfaces. It can be used to identify features such as ridges and valleys, and planar, convex, concave, or saddle shapes. Surfaces are segmented into regions based on these curvature features, and the segments and features then used for object recognition and registration.

The ability to compute curvature from meshes is complicated by the lack of an analytic definition for the surface shape. Meshes are defined at discrete vertices, while curvature is a function of how the surface behaves in a local region around the vertex. This is evident since curvature is based upon derivatives, which are themselves defined as a limit function. Thus, some assumptions on the behavior of the surface are required to estimate curvature for a localized set of vertices, such as a given vertex and its neighbors.

Past experience indicates that curvature metrics tend to be very sensitive to noise [23,19]. Scanners and sensors typically introduce some noise into the data. Small amounts of noise may be compensated for by smoothing, while large amounts may render the data unusable. Besides noise, the mesh resolution, i.e., how finely the surface is sampled, and regularity, i.e., the uniformity in size and shape of the mesh faces, also affect the accuracy of curvature estimates.

1.1. *Contribution*

In this work, we analyze curvature calculation methods applicable to triangular meshes. These fall into three categories: (1) fitting methods, (2) discrete estimation of curvature and curvature directions, and (3) estimation of a curvature tensor from which curvature and curvature directions can be found. We develop a process for evaluating the accuracy and stability of such methods using a suite of test cases that highlight the effects of mesh factors such as valence and mesh regularity, in addition to noise. We apply this suite to several existing algorithms and examine how reliably different algorithms predict the curvature values. Our evaluation process compares the error in mean, Gaussian, and principal curvatures, and the normal and principal curvature directions.

Knowledge of the accuracy, and sources of error, allows us to select algorithms that are robust and reliable for tasks such as shape matching and registration. An understanding of the errors in the curvature calculations can be combined with techniques from the Bayesian community to add confidence levels to the data, and to develop an understanding of when and why a method might break down.

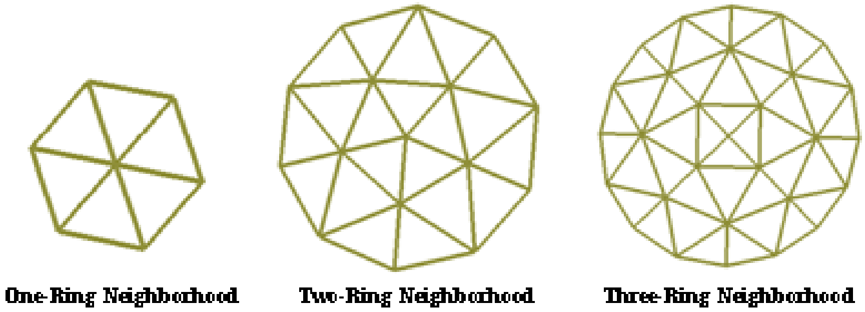

Fig. 1. Sample Test Case Meshes. Left: 1-Ring Neighborhood (valence=6), Middle: 2-Ring Neighborhood (valence=5), Right: 3-Ring Neighborhood (valence=4)

1.2. *Previous Work*

A number of researchers [14,30,1,5,20,18,10] have looked at curvature estimation from 3-D range images for computer vision applications. Range data provides a rectangular array of sample data, usually in the form of pixels. Many of the methods operate on an $N \times N$ window centered at a point, where N is an odd integer, typically 5 or 7. This window provides a natural orthogonal parameterization and well-defined diagonals. Mean and Gaussian curvature can be computed from first and second partial derivatives with respect to these preferred directions, or directly from the array of sample data. Methods that rely on this regular organization of data are not directly applicable for a general mesh.

Curvature estimation methods have also been developed specifically for meshes. Meshes have a more general structure than range images. Mesh representations have adjacency information embedded in the mesh connectivity, but without any regular organization or preferred direction.

We define neighbors as vertices that are part of the same face. All of the vertices that are neighbors to, i.e., share a common face with, a given vertex constitute its one-ring neighborhood. We extend this to a two-ring neighborhood by adding all of the neighbors of the one-ring vertices, and so on. Sample one-ring, two-ring, and three-ring meshes are shown in Figure 1. A given vertex of the mesh can have an arbitrary number of neighbors. These vertices need not be equidistant from the given vertex or equally spaced around the one-ring neighborhood.

Methods for 3-D range images that rely on the regular array structure, natural orthogonal parameterization, or preferred directions, are not readily adapted to mesh representations. However, methods that rely primarily on adjacency, such as surface fitting, may be adapted to mesh representations if a suitable set of vertices can be found. This set of vertices is typically an N-ring neighborhood, where commonly $N = 1$.

Past evaluations have compared specific methods, generally for very regular meshes, and have looked at the effect of noise and the benefits of smoothing [36,9,16]. The impact of other mesh factors has often been ignored. A few studies have compared a selection of curvature estimation methods, for example on range images [33,25], or meshes [34,16]. Others have focused on a particular method and then varied factors such as mesh resolution [36,4], the amount of noise added to the data [16], or the shape of the surface [17]. Most of these studies evaluate these methods for a very regular mesh. A few studies [37,16,29] also apply the methods to irregular meshes, but do not address the impact of the mesh irregularity.

A few papers have performed theoretical evaluations or experimental comparisons of selected curvature estimation methods. Meek and Walton [26] perform asymptotic analysis for several methods using both regular data (as in range data) and irregular data (as in meshes). The asymptotic behavior is important to insure that the methods would converge to the correct value, but as they state, the results may not be suitable for comparing different methods for fixed size meshes. While this asymptotic analysis was applied only to discretization and interpolation methods, Cazals and Pouget [4] note that 'interpolation fitting is always more ill-conditioned than approximation', so we might expect similar results for approximation techniques, such as least-squares fitting methods.

McIvor and Valkenburg [25], in comparing fitting methods for range data, note that there is bias in the curvature estimates since cylindrical and spherical patches cannot be represented exactly by a quadric. They also observe that for quadric fitting of surfaces with large curvature magnitude or with large sampling noise, the eigenvector associated with the surface normal may not have the smallest corresponding eigenvalue, causing the principal curvature and curvature direction estimation to break down. Overall, their results show the quadric fitting method performs better than the finite difference methods. Surazhsky et al. [34] compare several curvature estimation methods for meshes and conclude that a the Gauss-Bonnet scheme (angle deficit) provides the best Gaussian curvature estimate, and paraboloid (quadric) fitting is best for mean curvature estimations and second best for Gaussian curvature.

Overview: Section 2 discusses specific curvature estimation approaches for meshes. In Section 3, various curvature estimation methods are evaluated using our suite of test cases with parametric mesh perturbations and statistical analysis. Section 4 summarizes the conclusions of this study and outlines possible areas for future work.

2. Curvature Estimation

This section describes the methods that have been developed to calculate curvature on meshes. There are three basic approaches. Surface fitting involves finding an analytic function that fits the mesh locally. The curvature of the analytic function is well-defined [22]. Discrete methods develop either a direct approximation for the curvature, or an approximation of the curvature tensor, from which curvature and

Curvature Calculation Taxonomy - Fitting Methods								
						Princ	Crv	Req
Fit	Param	Paper	Data	Gauss	Mean	Crv	Dir	Norm
Range image methods								
Quadric	Grid	Flynn & Jain	NxN	X	X			
		Abdelmalek	NxN	X	X			
		Stokely & Wu	Voxels	X	X			
		McIvor & Valkenburg	Voxels	X	X		X	
Mesh methods								
Quadric	Planar Proj.	Hamann	1-Ring			X	X	
		Meek & Walton	1-Ring	X	X			
		Goldfeather & Interrante	1-Ring			X	X	
		Gatzke & Grimm	N-Ring			X	X	
Quadric	Natural	Gatzke & Grimm	N-Ring			X	X	
Cubic	Planar	Goldfeather & Interrante	1-Ring			X	X	X
Arb. Order	Planar	Cazals & Pouget	N Pts			X	X	
Conic	Implicit	Douros & Buxton	N Pts	X	X	X	X	
Radial Basis	Natural	Gatzke & Grimm	N-Ring			X	X	

Table 1. Curvature Calculation Taxonomy - Fitting Methods

curvature directions can be calculated. Discrete curvature equations are made from the continuous equations by approximating integrals as a summation of contributions attributed to each face or edge adjacent to a vertex.

2.1. *Fitting Methods*

The primary discriminator between fitting methods is the function chosen to model the local surface shape. Functions may be parametric, requiring a local parameterization of the surface near each vertex, or implicit. The chosen function is fit separately at each vertex of the mesh, with the method solving for the coefficients of the function. A local 3D coordinate frame, centered at the vertex, is useful for parameterization, and may also simplify estimation when using implicit functions.

The second discriminator is the number of vertices fit by the function. If too few vertices are fit, the problem is under-constrained. Therefore, a minimum number of vertices, based on the number of function coefficients, should be supplied. Fitting the minimum number of vertices defines an interpolating function where the function goes through each vertex. Fitting more than the minimum number of vertices leads to an approximating function, which minimizes some measure of distance from the function to the vertices, for example, a least-squares minimization. Fitting methods are listed in Table 2.1. This table indicates the type of data on which the algorithms operate, the parameterization method used, whether they require surface normals as input, and whether they compute Gaussian, mean, or principal curvature estimates, or principal curvature directions.

2.1.1. *Parameterization and Local Coordinates*

Many parameterization methods utilize a local 3D coordinate frame with its origin at the vertex. The normal vector at the vertex is frequently chosen as one axis of this frame. The vertex normal can be computed as the average of the face normals for the faces adjacent to the vertex, with various weightings applied, or as the normal to the plane that best fits the vertex and some number of nearby vertices. For methods that fit a surface to the data near the vertex, the normal can be replaced with the normal calculated from the surface fit. A local coordinate system is formed by the normal vector and two arbitrary orthogonal axes in a plane perpendicular to this vector. Transforming to such a local coordinate system does not restrict the curvature calculation, but does simplify the solution of the equations defining the surface representation.

One class of fitting methods represents the surface as a function of two parametric variables u and v in the form:

$$F(u, v) = (x(u, v), y(u, v), z(u, v))$$

The simplest representation is a height function, also referred to as a Monge patch. The height function is oriented relative to the local tangent plane, so that

$$F(u, v) = (u, v, f(u, v))$$

The parametric coordinates of the vertices are found by projecting the vertices onto the tangent plane. This projection can cause distortion in the relative distances between points, and the projection of complex regions can even cause folding. As an alternative, we can find a mapping that transforms the vertices to the plane while minimizing some measure of distortion. Several algorithms [13,8,31] have been developed to generate such mappings for a mesh that better preserve relationships and avoid folding.

2.1.2. *Quadric Fitting*

A popular choice for $f(u, v)$ is a quadratic function. Various forms of quadratic function have been fitted to range data [14,1,33,25] and to mesh representations [17,26,34]. For a general second-order polynomial with six coefficients, applied to a height function, we have:

$$z_i = f(u_i, v_i) = Au_i^2 + Bu_iv_i + Cv_i^2 + Du_i + Ev_i + F$$

where (u_i, v_i) is the parametric location of the i^{th} point in the tangent plane, and z_i is the height of the point above (or below) the tangent plane. Here, i runs from 1 to N, where N is the number of vertices being fit. The coefficients A through F are determined by solving a least-squares [28,12] problem. Two factors distinguish variations of this approach. First, the constant term, or the constant and linear terms, can be dropped. Dropping the constant term forces the fit to go through the vertex, while dropping the linear terms forces the normal to line up with the z axis of the local reference frame. The second factor is the number of vertices to include in the least-squares fit. One approach is to use just the vertices of the one-ring neighborhood. Alternatively, the neighborhood can be expanded to include a specified number of vertices in the least-squares fit. This larger number of vertices may be required based on the number of coefficients or to improve the stability of the solution. Cazals and Pouget [4] extend fitting to differential quantities of arbitrary order, using higher degree truncated Taylor expansions, called osculating jets.

2.1.3. *Cubic Fitting with Normals*

Goldfeather et al. [16] expand the quadric method by using a cubic fit of a system of equations formed from the coordinates and normal vectors at vertices on the one-ring neighborhood. Their focus is on calculation of principal curvature directions rather than the curvature magnitudes.

2.1.4. *Implicit Conic Functions*

Implicit functions provide an alternative fitting method that does not require a parameterization of the surface. Conic surfaces, particularly ellipsoids, have been used for surface fitting in applications such as medical imaging. Douros and Buxton [10] extend this approach to a general conic:

$$ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0$$

2.1.5. *Other*

Gatzke and Grimm[15] investigate a variation of the quadric fitting method using an alternate fitting function. In this variation of quadric fitting, they replace the projection to the tangent plane with a parameterization using the flattening algorithms of Desbrun et al.[8] and Sheffer[32]. This parameterization is intended to reduce distortion, and is less likely to produce folding. In addition to the quadratic equation fitting function, they explore using radial basis functions with a uniformly weighted Gaussian, which has well-behaved derivatives at the data points.

2.2. ***Discrete Methods***

One of the main motivations for discrete methods is to avoid the computational costs associated with fitting algorithms. These methods do not involve solving a least square problem and are very fast. However, many of these methods only provide a subset of gaussian, mean, and principal curvature directions (unlike a surface fit, from which any of this data can be calculated). Table 2.2 lists several common discrete curvature estimation methods.

Curvature Calculation Taxonomy - Discrete Operators							
					Princ	Crv	Req
Type	Paper	Data	Gauss	Mean	Crv	Dir	Norm
Range image methods							
Finite Diff.	McIvor & Valkenburg	NxN	X	X		X	
Srf Norm. Change	Flynn & Jain	NxN			X	X	X
Cross Patch	Stokely & Wu	NxN	X	X			
Mesh methods							
Meusnier-Euler	Chen & Schmitt	N Pairs			X	X	
	Hameiri & Shimshoni	N Pts			X	X	
Angle Deficit	Stokely & Wu	1-Ring	X				
	Meek & Walton	1-Ring	X				
	Meyer et al.	1-Ring	X				
Angle Excess	Stokely & Wu	1-Ring	X				
Integ. Abs. Mean	Dyn et al.	1-Ring		X			
Norm. Crv. Vec.	Meyer et al.	1-Ring		X		X	
Spherical Image	Meek & Walton	1-Ring	X				X

Table 2. Curvature Calculation Taxonomy - Discrete Operators

2.2.1. *Spherical Image*

The spherical image method [26] uses the unit normals of the one-ring vertices, translated to a common origin, to define a region of a unit sphere, and approximates Gaussian curvature as the ratio of the spherical area to the one-ring area.

2.2.2. *Angle Deficit*

The angle deficit method [33,26,27], based on the Gauss-Bonnet theorem, approximates Gaussian curvature as 2π minus the sum of the angles for the faces at a vertex, divided by an area associated with the vertex. Cohen-Steiner and Morvan [6] combine the angle deficit method with the integral absolute mean curvature form (below) from the theory of normal cycles, and present a bound on the error for a restricted Delaunay triangulation.

2.2.3. *Angle Excess*

The angle excess or turtle-walking method [33] is similar to the angle deficit method, but approximates Gaussian curvature as 2π minus the total turning angle for a path around a vertex divided by the area enclosed by the path. The path is taken as the boundary of a one-ring neighborhood.

2.2.4. *Integral of Absolute Mean Curvature*

The mean curvature is computed as a summation over the edges adjacent to a vertex of the angle between the faces adjacent to the edge multiplied by the edge length, and divided by four times the area associated with the vertex. Dyne et al. [11] pair this method with the angle deficit method to use as a cost function when optimizing the triangulation of a cloud of points.

2.2.5. *Meusnier and Euler Theorem*

Chen and Schmitt [5] estimate normal curvature and principal curvature directions by solving for the coefficients of the Dupin indicatrix [22] using three or more circular fits through a vertex and two of its neighbors. A normal section is the intersection of the surface with a plane containing the normal vector. Since there are many triples of points that can be used to create circular fits, the ones forming curves closest to a normal section are used. Hameiri and Shimshoni [18] use quadratic curves to estimate the normal section between the vertex with its normal and neighboring vertices.

2.2.6. *Curvature Normal Operator*

Meyer et al. [27] compute mean curvature by using a summation to approximate the integral of the Laplacian over the area associated with a vertex, and normalize by this area. This area can be a mixture of Voronoi and Barycentric area, depending on whether or not triangles are obtuse. They assume mild smoothness conditions and incorporate local operators to denoise arbitrary meshes while preserving features. The mean curvature is combined with Gaussian curvature computed using the angle deficit method to derive principal curvatures, and a least-squares method is employed to calculate principal directions.

2.2.7. *Derivative Calculation*

Csakany and Wallace [7] use a simplified approach to compute the second derivatives at a vertex of a mesh. They first compute the surface normal by averaging adjacent face normals. The normal defines the first partial derivatives. A substitution scheme is used to directly compute the second partial derivatives, which can be used to estimate curvature. Their scheme is considered a simplification of an auto-correlation method and a Hessian matrix method, and has been applied to both range images and tessellated data.

2.2.8. *Other*

Tang and Medioni [35] compute the sign and direction of curvature, without fitting or derivative calculation, using a voting scheme with weighting based on proximity. Their technique does not provide a curvature magnitude estimate.

Curvature Calculation Taxonomy - Curvature Tensor							
Type	Paper	Data	Gauss	Mean	Princ Crv	Crv Dir	Req Norm
Range image methods							
Integral Form.	Taubin	1-Ring			X	X	
	Hamieri & Shimshoni	N-Ring			X	X	X
Mesh methods							
Integral Form.	Taubin	1-Ring			X	X	
	Hamieri & Shimshoni	1-Ring			X	X	X
Per Face	Theisel et al.	1-Ring			X	X	X
	Rusinkiewicz	1-Ring			X	X	X

Table 3. Curvature Calculation Taxonomy - Curvature Tensor Estimation

2.3. *Estimating the Curvature Tensor*

Curvature tensor estimation is similar to the discrete methods, except that instead of estimating the curvature directly, a discrete estimation of the curvature tensor is created, and the curvatures and principal directions are calculated from the curvature tensor. These methods tend to have computational complexity lower than the fitting methods, but slightly higher than the discrete methods. Table 2.3 lists several curvature tensor estimation methods.

2.3.1. *Integral Formulation*

Taubin[36] proposes a method that estimates the tensor of curvature from the eigenvalues and eigenvectors of a 3 × 3 matrix, which approximates an integral as a summation around a one-ring neighborhood. He also incorporates a smoothing step for noisy meshes. A key benefit of his method is its simplicity, with the complexity being linear in both time and space. Hamieri and Shimshoni [18] propose modifications of Taubin's method by expanding to more points (primarily for range data), and weighting based on distance rather than triangle area, while Surazhsky et al. [34] proposed weighting based on angle. In the case of a general mesh, it is not clear whether variation in distance or angle will dominate.

2.3.2. *Per Face Tensor Calculation*

A recent approach calculates the curvature tensor separately for each face [37,29]. Given a face and the normal vectors at each vertex of the face, this curvature tensor is well-defined. To get the curvature tensor at a vertex, the tensors for each face adjacent to that vertex are averaged.

3. Evaluations

Previous studies do not provide an understanding of the differences between mesh size, regularity, and noise issues. Therefore, we develop a small number of tests to highlight both the detailed behavior of curvature estimation methods and a statistical analysis of errors.

The detailed behavior test case defines mesh parameters that distinguish between noise (perturbation normal to the surface) and triangulation effects (number, size, and regularity of triangles). We can then look at the error measures as we change parameter values. For example, we can empirically determine if the estimated curvature converges to the known value as the mesh cell size approaches zero. The detailed behavior test case uses an idealized (extremely regular) mesh, except for specific mesh parameter variations. This isolates the effect that specific mesh factors have on the curvature estimation, and provides insight into how sensitive different methods are to these factors.

The statistical analysis test case creates meshes containing vertices for a range of valences, with both regular and irregular mesh regions, and analyzes the errors with respect to the properties of these meshes. In practice, all of the detailed mesh parameters are likely to vary in a mesh. The statistical analysis helps us determine how the detailed behavior affects the overall behavior on a realistic mesh.

A suite of surface shapes, for which we know the exact curvature, is used to avoid the bias of methods that may be optimal for one particular surface shape. We use these tests to illustrate the behavior of several of the curvature estimation methods discussed above.

3.1. *Curvature Estimation Test Cases*

Test cases are constructed by first generating a mesh in the $X - Y$ plane, and then projecting the mesh in the Z direction onto surfaces of different shapes, represented by the following equations, as used previously by Hamann [17] and Cazals and Pouget [4]:

$$Sphere : x^2 + y^2 + z^2 = 4$$

$$Cylinder : x^2 + z^2 = 4$$

$$Ellipsoid : (x/3)^2 + (y/2)^2 + (z/4)^2 = 1$$

$$EllipticParaboloid : z = 2x^2 + y^2$$

$$Hyperboloid : z = 0.4(x^2 - y^2)$$

$$MonkeySaddle : z = 0.2(x^3 - 3xy^2)$$

$$CubicPolynomial : z = 0.15(x^3 + 2x^2y - xy + 2y^2)$$

$$TrigonometricFunction : z = 0.1[cos(\pi x) + cos(\pi y)]$$

$$ExponentialFunction : z = 0.1e^{2x+y-y^2}$$

3.1.1. *Detailed Mesh Parameters*

In order to assess the local curvature at a point on a surface, we project a planar triangular mesh onto the surface, centering the mesh at the target point. The mesh is a regular $N-$ring neighborhood, where $N \in\{1,2,3\}$. Sample two-ring meshes are shown in Figure 1. The vertices are equally spaced along the rings around the target vertex, except for variations of one of the seven parameters that are used to control the qualities of the mesh:

- n, the number of vertices (valence) in the first (adjacent) ring, with the second ring containing twice as many vertices,
- ϕ, the scale (a relative distance from the target vertex to the first ring of adjacent vertices, and between successive rings of vertices on the test surface),
- dR_T, the displacement of the target vertex normal to the surface,
- dR_A, the displacement of an adjacent vertex normal to the surface,
- $d\phi_T$, the displacement of the target vertex along the surface toward an adjacent vertex,
- $d\phi_A$, the displacement of an adjacent vertex along the surface toward or away from the target vertex, and
- $d\theta$, the displacement of an adjacent vertex along the surface toward a neighboring adjacent vertex.

The normal displacements, dR_T and dR_A, represent noise, *i.e.,* true deviation from the actual surface geometry, and are applied after the mesh is projected to the surface. $d\phi_T$, $d\phi_A$, and $d\theta$ represent perturbations of the triangulation. Examples of perturbations normal to and along the surface are shown in Figure 2. Moving the target point radially toward a point on the first ring, or moving a point of the first ring radially or circumferentially along the surface, reduces the regularity of the mesh.

We also added an offset and rotation to consider different target points and mesh orientations on the surface. This avoids bias that could occur if we looked only at special points, such as the points on the major and minor axes of an ellipsoid, or due to alignment of the mesh with the coordinate axes. For several of the algorithms tested, the accuracy at these special points was better than the accuracy of the method at a generic point on the surface.

The exact curvatures, normals, and principal directions are computed when the mesh is projected to the surface. For methods requiring surface normals, we can calculate approximate normal vectors or use the exact normals.

3.1.2. *Statistical Analysis Case*

For statistical analysis, we create a mesh containing 72 interior vertices (112 total), with valence ranging from three to ten, and containing both obtuse and non-obtuse triangles. This mesh is again created in the $X-Y$ plane and projected onto one of

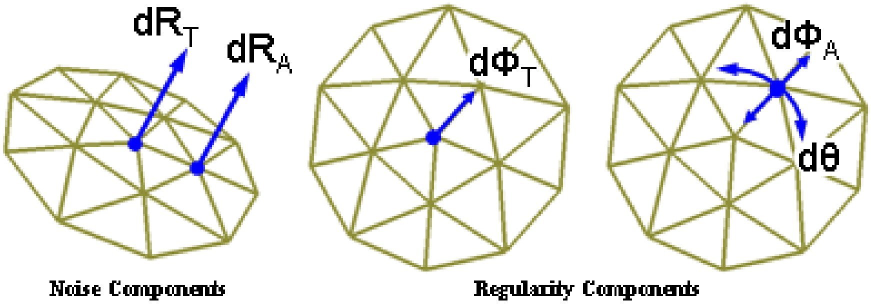

Fig. 2. Detailed Behavior Test Cases for a mesh projected onto a sphere. Left: Noise component normal to the surface at the target vertex (dR_T) and at an adjacent vertex (dR_A). Center: Mesh regularity component based om moving the target vertex away from center($d\phi_T$). Right: Mesh regularity components based on moving an adjacent vertex toward/away from the target ($d\phi_A$) or moving the adjacent vertex along the ring toward a neighboring adjacent vertex ($d\theta$).

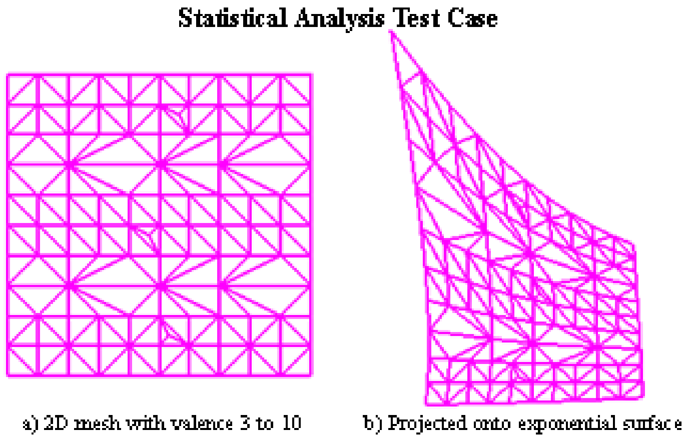
Statistical Analysis Test Case
a) 2D mesh with valence 3 to 10
b) Projected onto exponential surface

Fig. 3. Statistical Analysis Test Case. Left: The vertex layout in the $X-Y$ plane has valence ranging from three to ten (for interior vertices), and contains a mixture of obtuse and non-obtuse triangles. Right: The mesh projected to an exponential surface.

our surface shapes. Figure 3 shows an example of the statistical analysis mesh projected to the exponential surface. Statistics for curvature estimation can be broken down by (a) valence, (b) the presence or absence of obtuse angles at the vertex, or (c) the sign of the actual curvature. This breakdown is used to determine the impact of these three factors, and trends associated with the error in the curvature estimation.

3.2. *Experimental Results*

In the following sections, we present selected results for the three categories of curvature estimation methods.

3.2.1. *Curvature Estimation based On Fitting*

We look at the effect that various mesh parameters have on the error in the curvature estimate. The first factor that we consider is valence (i.e., the number of vertices making up the one-ring neighborhood around the target vertex) and its impact on the Gaussian curvature estimate. By comparing the error as the cell size decreases, we can plot the asymptotic behavior of the method. We will refer to this as the convergence of the method, and will consider the convergence to be good if the curve approaches the exact curvature value as the cell size approaches zero.

For one-ring neighborhoods with valences of three or four, the problem may be under-constrained, depending on the number of coefficients in the particular equation being fit. With enough vertices in the one-ring, or using multiple rings, the fitting methods are relatively insensitive to the valence. The cubic fit based on vertex locations and normals converges for all valences when using the exact surface normals, but has poor convergence when using normals calculated as the weighted average of the adjacent face normals.

As the cell size is decreased, corresponding to finer resolution, all of the fitting methods, except the cubic fit with calculated normals, converge to the correct value. Figure 4 illustrates the convergence for various fitting methods as a function of mesh resolution on a paraboloid. The conic fit performs well for several surface shapes, and as would be expected, is exact for the ellipsoid. However, the conic fit does not perform as well for some other surface types such as the exponential surface.

The biggest factor distinguishing performance for the fitting methods is the effect of noise normal to the surface, as shown in Figure 5. The quadric and conic fitting methods based on one-ring neighborhoods are extremely sensitive to this type of noise. The normals used with the cubic method effectively provide information from a second ring, and this was the most accurate fitting method in this situation. The fits based on two and three rings also performed well in the presence of noise normal to the surface, with a three-ring fit having no clear advantage over the two-ring fit. The Gaussian curvature estimates from the fitting methods were not particularly sensitive to varying the vertex location along the surface.

3.2.2. *Curvature Estimation Using Discrete Methods*

The impact of valence is most pronounced for the angle deficit method, as shown in Figure 6. This method converges to the exact value only for valence six. We should also note the distinction between point curvature, which we are using as our ground truth, and the integral of curvature over a region, upon which the angle deficit method is based. These methods will produce similar results if the curvature

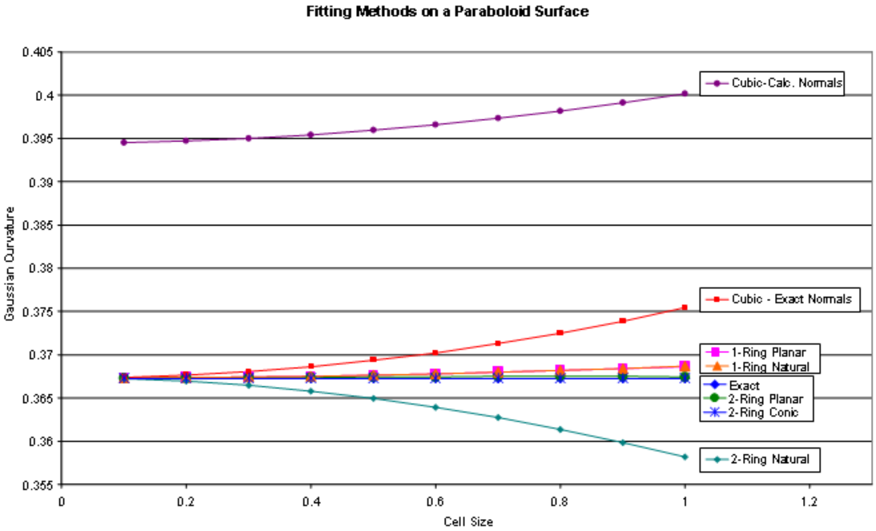

Fig. 4. Comparison of fitting methods applied to a paraboloid surface. A valence of six was used for the data shown. The cubic fit with computed normals does not converge to the exact curvature of 0.367. The one-ring and two-ring fits behaved similarly, with the one-ring fits being a little more accurate than the two-ring fits.

is relatively constant over the integration area, but may vary significantly in areas of rapidly changing curvature. There may be applications where one or the other type of curvature information is preferred, and this may lead to a different choice of methods. However, these valence plots show significant variations for essentially the same curvature region.

Like the one-ring fitting methods, the discrete curvature estimation methods [27] suffer from severe sensitivity to noise normal to the surface, as shown in Figure 5. But they are also very sensitive to perturbations of the mesh vertices along the surface, as compared to the fitting methods, as shown in Figure 7. This is likely caused by the reliance on angles and areas of the mesh faces, which do not enter directly into the fitting methods.

3.3. *Curvature Tensor Results*

Figure 8 illustrates Taubin's integral eigenvalue method [36] and Hameiri and Shimshoni's [18] modification of it on a sphere, calculated on a mesh centered at the origin, and on a mesh offset from the origin. Both meshes are projected in the Z direction to the same surface. Both methods match the exact curvature of the sphere for the mesh centered at the origin, but are less accurate for the offset mesh.

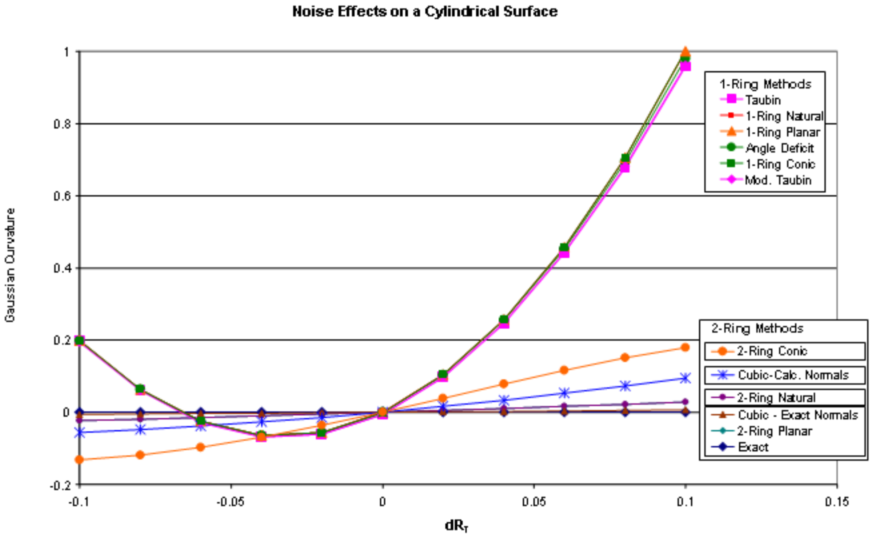

Fig. 5. Impact of noise normal to a cylindrical surface on the discrete and fitting methods. A valence of six was used for the data shown. The discrete methods and one-ring fitting methods exhibit extreme sensitivity to noise. The cubic fit behaves as a two-ring method and, along with the two-ring quadric fitting methods, shows the least sensitivity.

The main difference is that the projection of the offset mesh is not normal to the sphere, which degrades the mesh regularity. This effect is confirmed in Figure 9, where we start with a regular mesh, and move one of the adjacent points along the surface, generating errors from both methods. For the movement toward or away from the target vertex, the modified method performs better, but the modified method is more sensitive to movement around the ring, as shown in Figure 10. Being based on a one-ring neighborhood, they still suffer from severe sensitivity to noise normal to the surface.

3.3.1. *Statistical Analysis Results*

Results from the statistical analysis test case were generated for several surface shapes. They confirmed that the variations for some methods were very dependent on the type of surface. Overall, considering both Gaussian and mean curvature, the five most accurate methods were:

(1) The cubic fit with exact normals,
(2) The two-ring quadric planar fit,
(3) The two-ring conic fit,

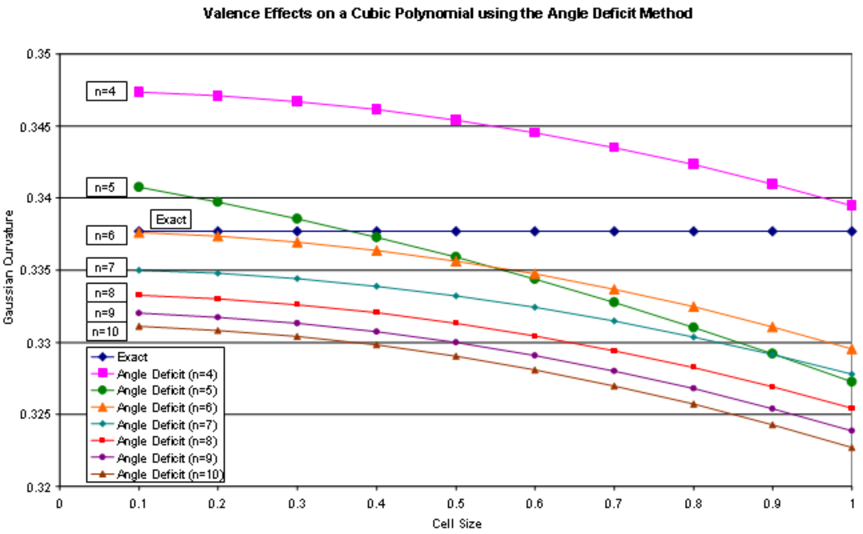

Fig. 6. Impact of valence on the accuracy of the angle deficit method, applied to a cubic polynomial. Increasing ϕ represents decreasing mesh resolution. Only valence six converges to the actual Gaussian curvature (0.338). This method is extremely inaccurate for a valence of three, probably due to the effect of obtuse triangles.

(4) The two-ring quadric natural fit, and
(5) The cubic fit with calculated normals,

The other methods had consistently larger error. The cubic fit using exact surface normals was best, with the two-ring fits (planar, natural, and conic) having similar errors. However, the two-ring conic fit and the cubic fit using calculated normals had much larger standard deviation than the other top methods. The cubic fit with exact normals and the planar and natural two-ring fitting methods were also most consistent across differences in valence, triangle shape, and curvature sign. Just looking at the overall errors can be deceptive, and mask deficiencies that are only uncovered by more detailed statistical analysis, or through the use of the specific noise analysis test cases. Results for the Gaussian and mean curvature calculations were similar. Mean curvature tends to be better behaved since it is an average rather than the product of the principal curvatures.

In order to place bounds on the accuracy of curvature estimates, it would be useful if methods could be identified that consistently over- or under-predict curvature magnitudes. In our evaluation, the cubic fit with calculated normals under-predicted the curvature magnitudes in most cases, and the two-ring conic fit predictions were larger (more positive or less negative) than the actual Gaussian curvature. However,

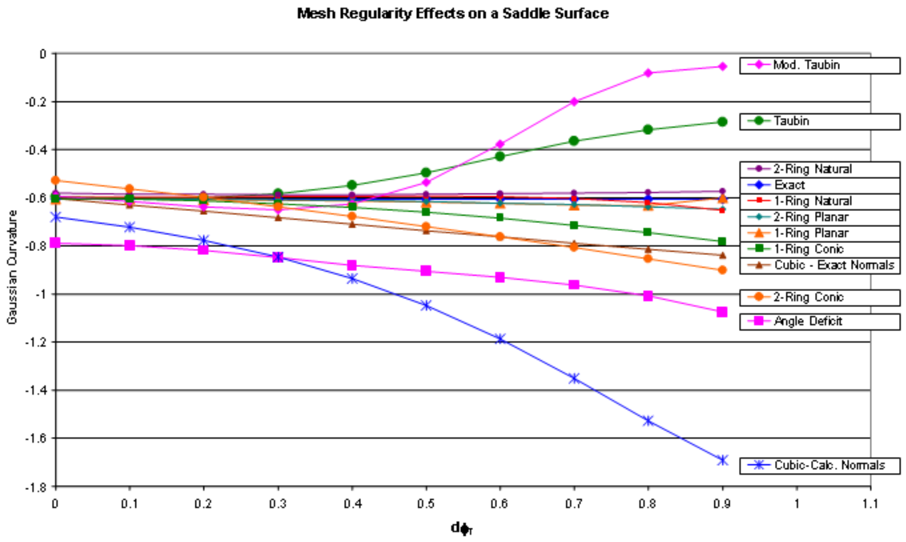

Fig. 7. Impact of moving the target vertex along the surface toward a one-ring point ($d\phi_T$). The valence is six for all methods. The discrete methods and the cubic fit with computed normals are very susceptible to this mesh quality issue, while the other methods show little sensitivity.

as discussed above, these methods had other problems that limit the application of these trends. The cubic fit using exact normals and the two-ring quadric fitting methods had smaller error magnitudes, but the sign of the error did not exhibit a consistent trend across the set of test shapes.

3.4. *Discussion of Results*

The accuracy for the conic fitting method was very dependent on the type of surface being fit. This points to the importance of comparing methods for more than one type of surface. If an evaluation case is based on the same equation as the fitting method, the results of the evaluation will not necessarily reflect performance for other surfaces to which the method will be applied.

The accuracy of fitting methods and the angle deficit method have been demonstrated in previous studies, as mentioned in Section 1.2. Our analysis confirms the benefits of fitting methods, but identifies deficiencies in the angle deficit method, and conic fitting methods. Even without noise, the statistical analysis indicates that the two-ring fitting is superior to the one-ring fitting methods. The detailed behavior suggests that noise normal to the surface severely degrades the one-ring methods, which have higher noise sensitivity.

The cubic fit appears promising, but the sensitivity to the calculation of the nor-

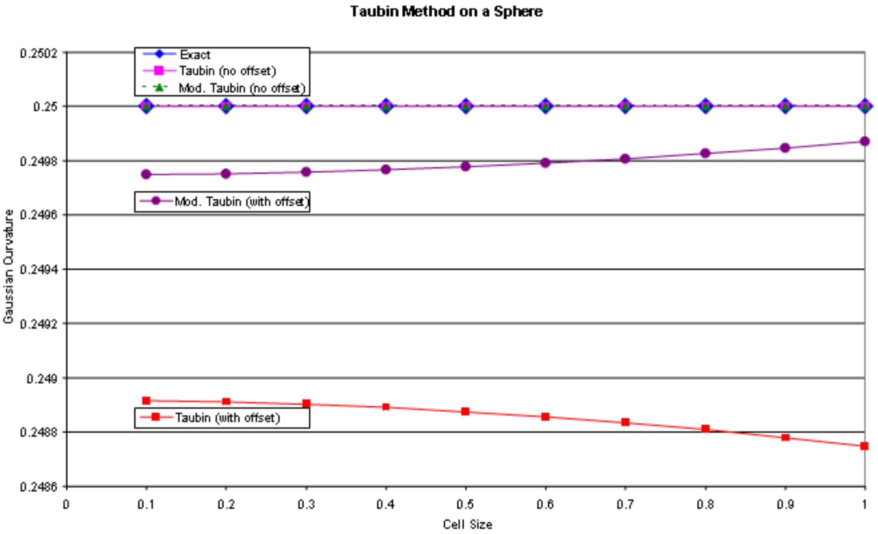

Fig. 8. Integral Eigenvalue method applied to a sphere as a function of cell size. A valence of six is used for the data shown. The methods match well for a regular mesh centered at the origin, but degrade due to the minor distortion from projection of an offset mesh.

mals is a severe drawback. The principal curvature direction calculations appeared more stable and less sensitive to mesh regularity than the curvature magnitudes. See Goldfeather[16] for further discussion comparing calculation of principal directions.

The discrete curvature methods are appealing because of their speed. Fitting is by its nature a more expensive computation. However, the sensitivity to valence, noise, and mesh regularity limit the usefulness of the discrete curvature estimates to very regular meshes for which either noise is absent or smoothing has been applied. The authors of these methods have proposed applying smoothing algorithms for cases with noise. But smoothing can also mask surface detail if not applied judiciously.

The fitting methods based on two or more rings have better overall performance, albeit at a greater computational cost. In our tests, accuracies for three ring neighborhoods did not warrant the increased cost due to the size of the fitting problem, so a surface fit based on a two ring neighborhood is recommended.

Conic fitting is usually phrased as a least-squares solution that minimizes $F(x, y, z)^2$. Scaling the conic equation by a constant value does not change the zero set, but it does change the value of $F(x, y, z)$. For this reason, we have found fitting to be more stable if the points are first transformed to a local coordinate system centered around the origin, with the normal pointing in the y direction.

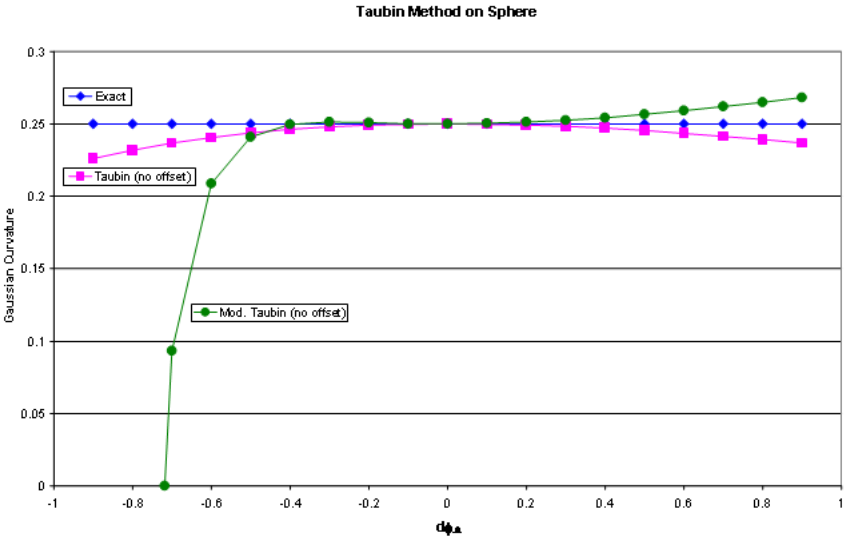

Fig. 9. Effect of perturbation of an adjacent vertex along the surface toward or away from the target vertex. Both variation of the Integral Eigenvalue method show sensitivity to the mesh regularity.

Fitting using radial basis functions did not yield suitable curvature estimates. However, there are a variety of possible formulations that may be worth investigating.

Generating the parameterization by projecting to the tangent plane is quick and easy. Alternative parameterization, based on a flattening [8] of the local mesh, avoids potential problems due to folding or distortion when the mesh is projected to a plane. However, these techniques require more work and do not provide much accuracy improvement. The behavior for a two-ring fit parameterized by a natural flattening technique were similar to the two-ring planar fit.

These results demonstrate the value of our analysis methods to uncover the detailed behavior of curvature calculation methods on triangular meshes, and an approach to statistical analysis that can provide practical assessment of new or existing methods. Source code and data files for the metrics and methods used in this analysis are available from `www.cs.wustl.edu/MediaAndMachines/Curvature`.

4. Conclusions

We have presented a suite of test cases that model mesh variations to assess the impact of mesh resolution, regularity, valence, and noise on the accuracy of curvature calculation algorithms for triangular meshes. In addition to fundamental mesh

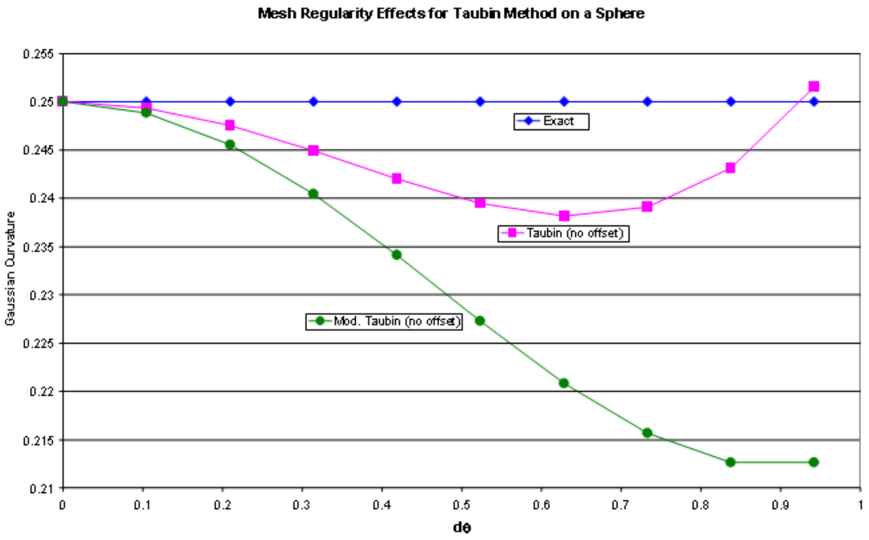

Fig. 10. Perturbation introducing non-uniform spacing around the ring. Again, both methods exhibit sensitivity to this mesh regularity parameter.

issues, this suite includes statistical analysis that also addresses different aspects of curvature estimation error. We have provided a summary of existing curvature estimation methods and have applied our evaluation suite to the most common surface fitting and discrete methods, to produce guidelines for choosing an algorithm.

Further work will investigate if the behavior of curvature estimation methods, based on mesh resolution and other factors, can be used to place bounds on the error in the curvature estimates.

References

1. Nabih N. Abdelmalek. Algebraic arror analysis for surface curvatures and segmentation of 3-d range images. *Pattern Recognition, Vol. 23*, pages 807–817, September 1989.
2. N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *To appear in 16th ACM Sympos. Comput. Geom.*, July 2000.
3. F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, August 1999.
4. F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 177–187. Eurographics Association, 2003.
5. X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In

Proceedings of the European Conference on Computer Vision (ECCV), pages 739–743, 1992.

6. David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321. ACM Press, 2003.
7. P. Csakany and A.M. Wallace. Computation of local differential properties on irregular meshes. In *Proc. of IMA Conference on Mathematics of Surfaces (NIPS 2000)*, pages 19–33, Cardiff, 2000.
8. M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Eurographics 2002, Vol. 21, Number 2*, 2002.
9. M. Desbrun, M. Meyer, P. Schroder, and A. Barr. Discrete differential-geometry operators in nd, 2000.
10. I. Douros and B. Buxton. Three-dimensional surface curvature estimation using quadric surface patches. *Scanning 2002 Proceedings, Paris*, May 2002.
11. Nira Dyn, Kai Hormann, Sun-Jeong Kim, and David Levin. Optimizing 3d triangulations using discrete curvature analysis. pages 135–146, 2001.
12. Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
13. M. S. Floater. Parameterization and smooth approximation of surface triangulations. In *Computer Aided Geometric Design 14*, pages 231–250, 1997.
14. P. J. Flynn and A. K. Jain. On reliable curvature estimation. *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 110–116, June 1989.
15. T. D. Gatzke and C. M. Grimm. Assessing curvature metrics on triangular meshes. Technical report, Washington University, St. Louis Missouri, jun 2003.
16. Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004.
17. Bernd Hamann. Curvature approximation for triangulated surfaces. In Gerald E. Farin, Hans Hagen, Hartmut Noltemeier, and Walter Knödel, editors, *Geometric Modelling*, volume 8 of *Computing Supplement*, pages 139–153. Springer, 1993.
18. E. Hameiri and I. Shimshoni. Estimating the principal curvatures and the darboux frame from real 3d range data. In *First International Symposium on 3D Data Processing Visualization and Transmission*, pages 258– 267, 2002.
19. A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 517–526. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
20. A. Hilton, J. Illingworth, and T. Windeatt. Statistics of surface curvature estimates. *Pattern Recognition, Vol 28*, pages 1201–1221, February 1995.
21. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics (SIGGRAPH '92 Proceedings), Vol. 26, No. 2*, pages 71–78, July 1992.
22. Josef Hoschek, Dieter Lasser, and Larry L. Schumaker. *Fundamentals of computer aided geometric design.* A. K. Peters, Ltd., 1993.
23. Victoria L. Interrante. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. *Computer Graphics*, 31(Annual Conference Series):109–116, 1997.
24. S. K. Lodha and R. Franke. Scattered data techniques for surfaces. *Scientific Visualization Dagstuhl '97, Hagen, Nielson, and Post (eds)*, pages 181–222, August 1997.
25. Alan McIvor and Robert J. Valkenburg. A comparison of local surface geometry estimation methods. *Machine Vision and Applications*, 10(1):17–26, 1997.

26. D. S. Meek and D. J. Walton. On surface normal and gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design 17*, pages 521–543, February 2000.
27. M. Meyer, H. Lee, A. H. Barr, and M. Desbrun. Generalyzed barycentric coordinates for irregular n-gons. In *Journal Of Graphic Tools*, 2002.
28. Vaughan Pratt. Direct least-squares fitting of algebraic surfaces. *Computer Graphics*, 21(4):145–152, 1987.
29. Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004)*, pages 486– 493, September 2004.
30. Peter T. Sander. Generic curvature features from 3-d images. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 1623–1635, November 1989.
31. A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. In *Engineering with Computers, 17 (3)*, pages 326–337, 2001.
32. Alla Sheffer and Eric De Sturler. Smoothing an overlay grid to minimize linear distortion in texture mapping, 2002.
33. E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-d objects: Five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 833–839, August 1992.
34. Tatiana Surazhsky, Evgeny Magid, Octavian Soldea, Gershon Elber, and Ehud Rivlin. A comparison of gaussian and mean curvatures triangular meshes. *To appear in 2003 IEEE International Automation (ICRA2003)*, May 2003.
35. Chi-Keung Tang and Gerard G. Medioni. Robust estimation of curvature information from noisy 3d data for shape description. In *ICCV (1)*, pages 426–433, 1999.
36. G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. *5th Intl. Conf. on Computer Vision (ICCV'95)*, pages 902–907, June 1995.
37. Holger Theisel, Christian Rossl, Rhaleb Zayer, and Hans-Peter Seidel. Normal based estimation of the curvature tensor for triangular meshes. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pages 288–297. IEEE Computer Society, 2004.