Estimating Curvature on Triangular Meshes

T.D. Gatzke, C.M. Grimm

Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A

Abstract

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 the sensitivity of curvature calculations to noise, mesh resolution, and mesh regularity. These tests are applied to existing discrete curvature approximation techniques and common surface fitting methods. We 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.

Key words:
Computational Geometry, Object Modeling, Curve, Surface, Solid, Object Representations

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 [Amenta et al., 2000, Bernardini et al., 1999, Hoppe et al., 1992, Lodha and Franke, 1997], 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 can be segmented into regions based on these curvature features, and the segments and features can be 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 assumption on the behavior of the surface is required to estimate curvature for a localized set of vertices. Past experience indicates that curvature metrics tend to be noisy [Interrante, 1997, Hertzmann and Zorin, 2000]. 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. Other factors include the resolution, i.e., how finely the surface is sampled, and regularity, i.e., the uniformity in size and shape of the mesh faces.

1.1 Mesh Representation vs. Range Data

A number of researchers [Flynn and Jain, 1989, Sander, 1989, Abdelmalek, 1989, Stokely and Wu, 1992, Hilton et al., 1995] have looked at curvature estimation from 3-D range images for computer vision applications. While some of these methods may be adapted to mesh representations, others cannot. Range data provides a rectangular array of sample data, usually in the form of pixels. Adjacency is implicit in this array structure. Many of the methods operate on an $N \times N$ window centered at a point, where the parameter N is an odd integer, typically 5 or 7. This window provides a natural orthogonal parameterization that can be used to compute first and second partial derivatives with respect to the parametric coordinates. Mean and Gaussian curvature can be computed from these derivatives. The regular nature of the data also provides preferred directions. A curve through points aligned along the horizontal, vertical, or either of the two diagonals can provide derivatives and curvatures in these directions. Other approaches use the array of sample data directly to estimate curvature.

Mesh representations have adjacency information embedded in the mesh connectivity, but there is not generally 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 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

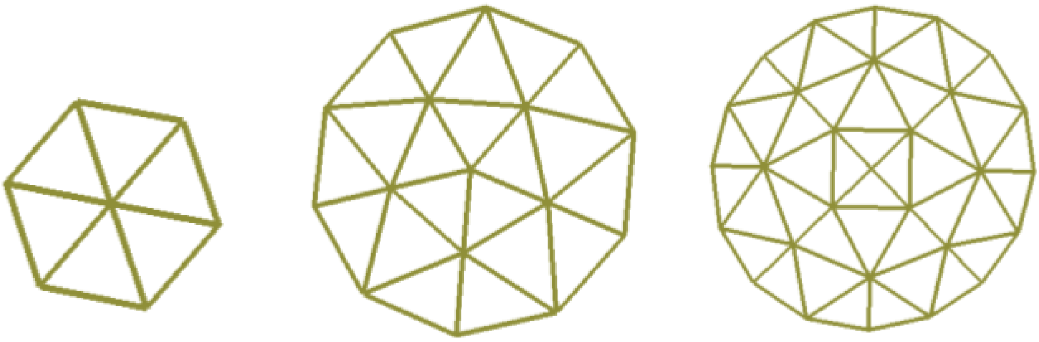

Fig. 1. Sample Test Case Meshes. Left: 1-Ring Neighborhood (valence=6), Middle: 2-Ring Neighborhood (valence=5), Right: 3-Ring Neighborhood (valence=4)

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. There is no guarantee that vertices in the i^{th} ring are closer than vertices in the j^{th} ring for $i < j$.

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 most commonly $N = 1$.

1.2 Surface Fitting vs. Discrete Methods

Curvature calculation methods fall into one of two main categories. The first category, surface fitting, involves finding a function that fits the mesh locally, and calculating the curvature of the fitting function. This function can be interpolating, where the function goes through each vertex, or approximating, where the function minimizes some measure of distance from the vertices. In both cases, the method solves for the coefficients of the function. Interpolating functions require a specific relation between the number of vertices and the number of coefficients in the function. Approximating functions require a minimum number of vertices and compute the coefficients as a solution to a least square minimization.

The second category, discrete methods, involves developing discrete approximations based on the definition of curvature. These methods do not use an intermediate analytical fit of the surface. Discrete methods often approximate an integral equation around a vertex by a summation of contributions attributed to each face or edge adjacent to the vertex.

1.3 Contribution

We expect that certain methods may consistently outperform others for real meshes with irregular distributions of vertices and some amount of noise. For example, it is reasonable to expect an algorithm based on a bigger support region (more vertices) to perform better in the presence of noise.

Past evaluations have compared specific methods, generally for very regular meshes, and have looked at the effect of noise and the benefits of smoothing. The impact of other mesh factors has often been ignored. In addition to presenting a survey of curvature calculation methods, we develop a suite of test cases to evaluate the accuracy of these methods. We construct these test cases using mesh regularity and noise parameters and assess their effect on curvature estimation accuracy. We apply this suite to the existing algorithms and examine how reliably different algorithms over- or under-predict the actual curvature values.

In this paper we:

- Identify methods for which we can establish bounds on accuracy.
- Determine which method is likely to perform best in a particular scenario.
- Develop an understanding of when and why a method might break down.

This knowledge allows us to select algorithms that are robust and reliable for application to tasks such as shape matching and registration. An understanding of the errors in the curvature calculations can be combined with techniques from the Baysian community to add confidence levels to the data.

Curvature metrics include scalar properties such as maximum and minimum principal curvatures, mean and Gaussian curvatures, and vector quantities such as principal curvature directions. In this paper we focus on the estimation of curvature magnitudes and principal directions, and discuss surface normals only to the extent that they affect the curvature calculation. We assume meshes contain only triangular faces. This is the simplest and most common representation and is not really a restriction since meshes containing higher-order faces can always be decomposed into sets of triangles.

Section 2 highlights some previous work on curvature calculations for meshes. In Section 3, various curvature estimation methods are evaluated using our suite of test surfaces with parametric mesh perturbations. Section 4 summarizes the conclusions of this study and outlines possible areas for future work.

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		Goldfeather & Interrante	1-Ring			X	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

2 Curvature Estimation

This section describes the methods that have been developed to calculate curvature on meshes. These methods are again categorized as fitting methods or discrete methods.

2.1 Fitting Methods

Fitting methods are listed in Table 2.1. Fitting methods depend first of all on the function chosen to model the local surface shape, and may also require a local frame of reference and surface parameterization.

2.1.1 Parameterization and Local Coordinates

When calculating curvature at a vertex, many methods utilize a local reference frame centered at the vertex. This reference frame is based on the normal vector at the vertex. This normal can be approximated by the normal to the plane that best fits the vertex and some number of nearby vertices. Other approximations are based on the average of the face normals for the faces adjacent to the vertex, with various weightings applied. For methods that fit a surface to the data near the vertex, the normal can be refined using 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. Using a standardized local coordinate system does not restrict the curvature calculation, but does simplify the solution of the equations defining the surface representation.

Most fitting methods represent 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 as a height function, also referred to as a Monge patch, relative to the local tangent plane, so that

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

and the parametric coordinates are the projections of vertices onto the tangent plane. This projection can cause distortion in the relative distances between points, and 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 [Floater, 1997, Desbrun et al., 2002, Sheffer and de Sturler, 2001] have been developed to generate such mappings for a mesh that better preserves relationships and avoids folding.

2.1.2 Quadric Fitting

Locally fitting a quadric function is the most popular curvature estimation technique, both for range data [Flynn and Jain, 1989, Abdelmalek, 1989, Stokely and Wu, 1992, McIvor and Valkenburg, 1997] and for mesh representations [Hamann, 1993, Meek and Walton, 2000]. For a general second-order polynomial with six coefficients, applied to a height function, we have:

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

The coefficients are determined by solving a least squares problem. Two factors distinguish variations of this approach. First, the constant term, or the constant and linear terms, can be dropped. The former forces the fit to go through the vertex, while the latter forces the normal to line up with the z axis of the local reference frame. The second factor is the choice of vertices to fit. This generally corresponds to the one-ring neighborhood, but can be expanded to a larger neighborhood [Gatzke and Grimm, 2003].

2.1.3 Cubic Fitting with Normals

Goldfeather and Interrante [2004] expand the quadric method by adding third-order terms, and use the coordinates and normal vectors at vertices on the one-ring neighborhood to set up a system of equations. Their focus is on calculation of principal curvature directions rather than the curvature magnitudes.

2.1.4 Implicit Conic Functions

Conic surfaces, particularly ellipsoids, have been used for surface fitting in applications such as medical imaging. Douros and Buxton [2002] extend this approach to a general conic:

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

Unlike other surface fitting techniques, this method does not require a parameterization.

2.1.5 Other

Gatzke and Grimm [2003] investigate use of radial basis functions as an alternative to the quadric polynomial. These functions are implemented for both the planar projection and parameterization using the flattening algorithms of Desbrun et al. [2002], Sheffer and Sturler [2002]. They choose a uniformly weighted Gaussian so that the derivatives at the data points are well-behaved.

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 generally very fast. 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							
Integral Form.	Taubin	1-Ring			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				
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 [Meek and Walton, 2000] uses the unit normals for 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 [Stokely and Wu, 1992, Meek and Walton, 2000, Meyer et al., 2002] 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.

2.2.3 Angle Excess

The angle excess or turtle-walking method [Stokely and Wu, 1992] 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 Formulation

Taubin [1995] 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.

2.2.5 Curvature Normal Operator

Meyer et al. [2002] 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 square method is employed to calculate principal directions.

2.2.6 Derivative Calculation

Csakany and Wallace [2000] 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 an Hessian matrix method which have been applied to both range images and tessellated data.

3 Evaluations

Stokely and Wu [1992] look at the effect of patch size on range images for spheres of various sizes, with and without added noise, and on image data from the left ventricle of a human heart. As the patch size decreases, accuracy decreases since the digitization error dominates. They also note that as the patch size increases, the parabolic fit over-predicts the curvature of a sphere.

Hamann [1993] uses surfaces defined by eight different types of analytic functions to assess the accuracy of the quadric fitting algorithm. His analysis is based on a very regular mesh and does not consider noise in the data.

Taubin [1995] looks at the effect of mesh resolution using regular meshes for a sphere, torus, and a more general analytic surface shape. He also applies his method to different triangulations (although with similar resolution and uniformity) to determine the effect of the triangulation on the error.

McIvor and Valkenburg [1997] apply quadric fitting and finite differences to range image data from planar, cylindrical, and spherical surfaces with Gaussian noise added. They note that there is bias in the curvature estimates since cylindrical and spherical patches cannot be represented exactly by a quadric. Also, large curvature magnitude or curvature variation over the patch size can cause the selection of an incorrect eigenvector and the method breaks down. Their results show the quadric fitting method performs better than the finite difference methods.

Meek and Walton [2000] perform asymptotic analysis for several methods using both regular data (as in range data) and irregular data (as in meshes). They state that asymptotic analysis applies only to discretization and interpolation methods, but not to least-square fitting methods. Since asymptotic analysis predicts behavior as the cell size h approach zero, they note that if h is not close enough to zero, the results may not be suitable for comparing different methods.

Goldfeather and Interrante [2004] create an analytic surface for comparing their cubic method to quadric fitting. They also introduce some randomness into the mesh on the surface to look at the affect of mesh irregularity. As mentioned previously, their focus is on the calculation of principal directions rather than curvature magnitudes.

Each of these studies has concentrated on a few specific issues related to curvature estimation. Meek and Walton [2000] highlight some of the advantages of asymptotic analysis, but two major shortcomings are (a) not being applicable to least squares quadric fitting, which is the most common approach, and (b) the fact that we generally have a fixed resolution mesh, so behavior as the cell size goes to zero is not so significant. Neither have these studies developed an understanding of the differences between mesh size, regularity, and noise issues. Therefore, we develop a small number of tests using surfaces for which we know the exact curvature. We assess how noise (perturbation normal to the surface) and triangulation effects (number, size, and regularity of triangles) impact the accuracy of the curvature calculations. We apply these tests to the majority of the curvature estimation methods for meshes discussed above.

3.1 Noise Analysis Test Cases

We take a systematic look at the effect of different components of noise on the curvature calculations. For space reasons we focus on the Gaussian curvature and curvature directions. The test cases are also suitable for evaluating mean and principal curvatures. Source code and data files for all metrics and methods are available from `www.cs.wustl.edu/MediaAndMachines/Curvature`.

Since the curvature calculation is a local operation on a mesh, we use simple shapes to model the range of positive and negative Gaussian curvatures. We use portions of three basic shapes as our test cases: a sphere (constant curvature), an ellipsoid (positive Gaussian curvature), and a saddle surface (negative Gaussian curvature). Geometric parameters are the radius r for the sphere, and two additional parameters, a and b, for the ellipsoid and saddle shapes.

The sphere, ellipsoid, and saddle equations are:

$$(x/r)^2 + (y/r)^2 + (z/r)^2 = 1$$

$$(x/r)^2 + (y/a)^2 + (z/b)^2 = 1$$

$$x - (a - \sqrt{a^2 - y^2}) + (b - \sqrt{b^2 - z^2}) = 0$$

For the sphere we use $r = 1.0$ which results in a Gaussian curvature value of 1.0. For the ellipsoid we use $r = 1.0$, $a = 1.0$, $b = 0.5$, which results in a Gaussian curvature value of 4.0. For the saddle surface we use $r = 1.0$, $a = 1.0$, $b = 0.5$ which results in a Gaussian curvature value of -2.0.

The test cases built from these surfaces are split into three types: those that have points on the surface but with perturbations of the triangulation, those with noise in the normal direction, and different mesh resolutions.

3.1.1 Mesh Parameters

In order to assess the local curvature, we generate a triangular mesh around a target point on the surface, encompassing an $N-$ring neighborhood around a point as described previously, where $N \in \{1,2,3\}$.

For methods requiring surface normals, we can calculate approximate normal vectors or use the exact normals. To calculate approximate normal vectors at the target point and its $N-1$-ring neighbors, N must be greater than 1.

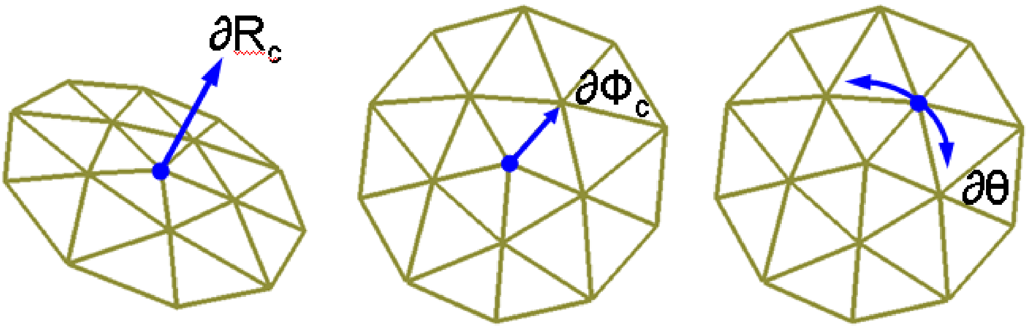

Fig. 2. Mesh Noise Components. Left: Target point moved normal to the surface, Center: Target point moved away from center, Right: Adjacent point moved along ring.

Parameters that control the qualities of the mesh include:

- n, the number of points in the first ring, with the second ring containing twice as many points, and
- ϕ, the distance from the target point to the first ring, and between successive rings, measured as an angle at the center of the sphere.

The vertices are equally spaced along the rings around the target vertex, except for noise perturbations described below.

The ellipsoid and saddle meshes were created from the spherical mesh by projecting points along the X axis to the surface defined by one of the equations above. Sample two-ring meshes are shown in Figure 1. The mesh is also rotated so that the curvature directions are not aligned with the coordinate axes.

3.1.2 Noise Parameters

We define five components of noise, two of which are applied to the target vertex, and three of which are applied to a vertex in the first ring. Examples of perturbations normal to and along the surface are shown in Figure 2. The target vertex can be perturbed either in a direction normal to the surface (left), or along the surface (center) toward one of the vertices of the first ring. Similarly, a vertex in the first ring can be perturbed in the normal direction, on the surface toward or away from the central target vertex, or on the surface along the one-ring toward a neighbor on the same ring (right) while maintaining a constant distance from the target vertex.

The perturbation component normal to the surface represents noise, *i.e.,* a true deviation from the actual surface geometry. The components along the surface, radial or circumferential, do not deviate from the true geometry, but rather represent the effects of mesh quality. The baseline two-ring neighborhood around the target point is very regular, with fairly uniform angles and edge lengths. 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 this regularity. The perturbations are limited to two vertices, applied to one vertex at a time to track the effects of the specific perturbation. As a result, the region around the target vertex is still, in general, better behaved than that of a general mesh, which can have angle and edge length variations associated with each adjacent vertex.

We can extend these noise-generation techniques to a complete mesh representing an analytic surface. The magnitude of the noise is specified based on a fraction of the smallest triangle edge around a vertex. We keep the magnitude of the noise below 50% to avoid (as much as possible) folding in the mesh. For each vertex we can either slide it along the surface (changing the mesh quality) or off the surface (introducing noise).

3.2 Fitting Results

The first factor that we look at is valence, which is the number of vertices making up the one-ring neighborhood around the target vertex, and its impact on the Gaussian curvature estimate. All of the fitting methods that use just the locations of the vertices in the one-ring neighborhood perform very poorly if there are less than four vertices in the one-ring and most perform poorly if there are less than five vertices. In this case, depending on the number of coefficients in the particular equation being fit, the problem may be underconstrained. With more vertices in the one-ring or 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 based on the exact surface normals, but has poor convergence when using normals calculated as the weighted average of the adjacent face normals.

For a regular mesh, all of the fitting methods converge to the correct value as the cell size is decreased, corresponding to finer resolution. Figure 3 illustrates the convergence for various methods as a function of mesh resolution. Except for a valence of three, the quadric fit methods converge to positive values from above, and to negative values from below, i.e., they tend to over predict the curvature magnitude. The cubic method converges from below for a valence of three and from above for all other valences. The conic fit converges from below for the saddle surface, and as would be expected, is exact for the ellipsoid. 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.

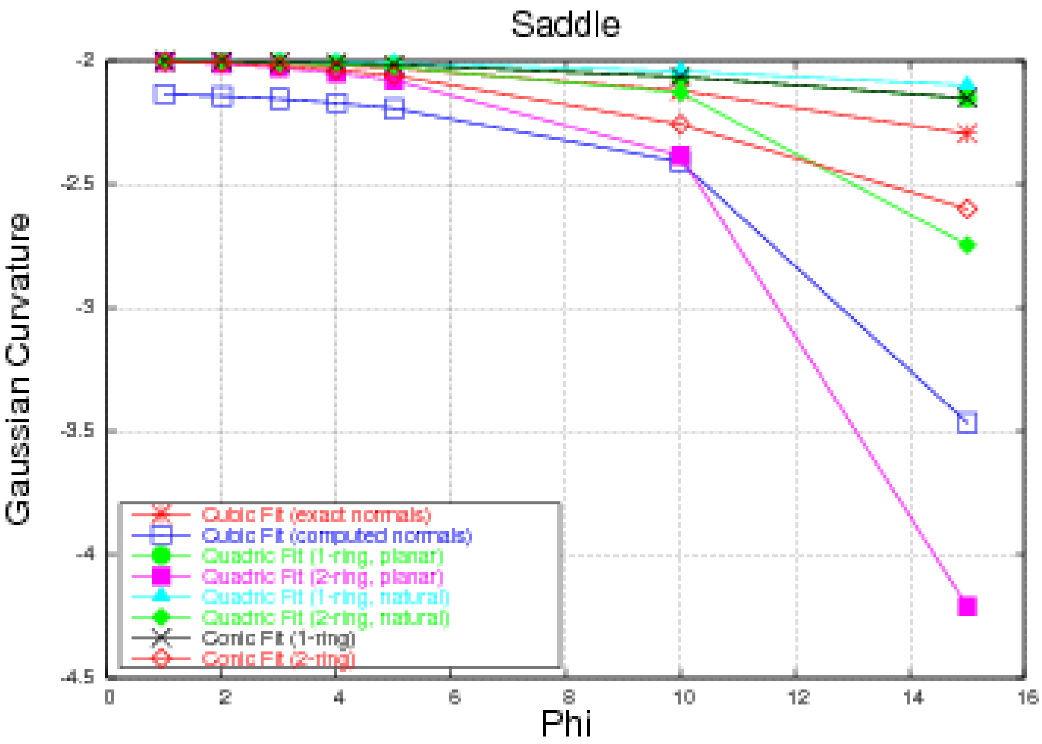

Fig. 3. Comparison of fitting methods applied to a saddle surface. A valence of six was used for the data shown. The cubic fit with computed normals does not converge to the exact curvature of -2.0. The one-ring and two-ring fits behaved similarly, with the one-ring fits being a little more accurate than the two-ring fits.

The biggest factor distinguishing performance for the fitting methods is the effect of noise normal to the surface, as shown in Figure 4. 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.

3.3 Discrete Results

The integral eigenvalue method performs poorly for valence three and four. The impact of valence is most pronounced for the discrete curvature operator based on the angle deficit method, as shown in Figure 5. This method is very poor for valence three, converges to the correct value for valence four and six, and is marginal for other values. The integral eigenvalue method is biased in the negative direction for all cases, while the angle deficit method under

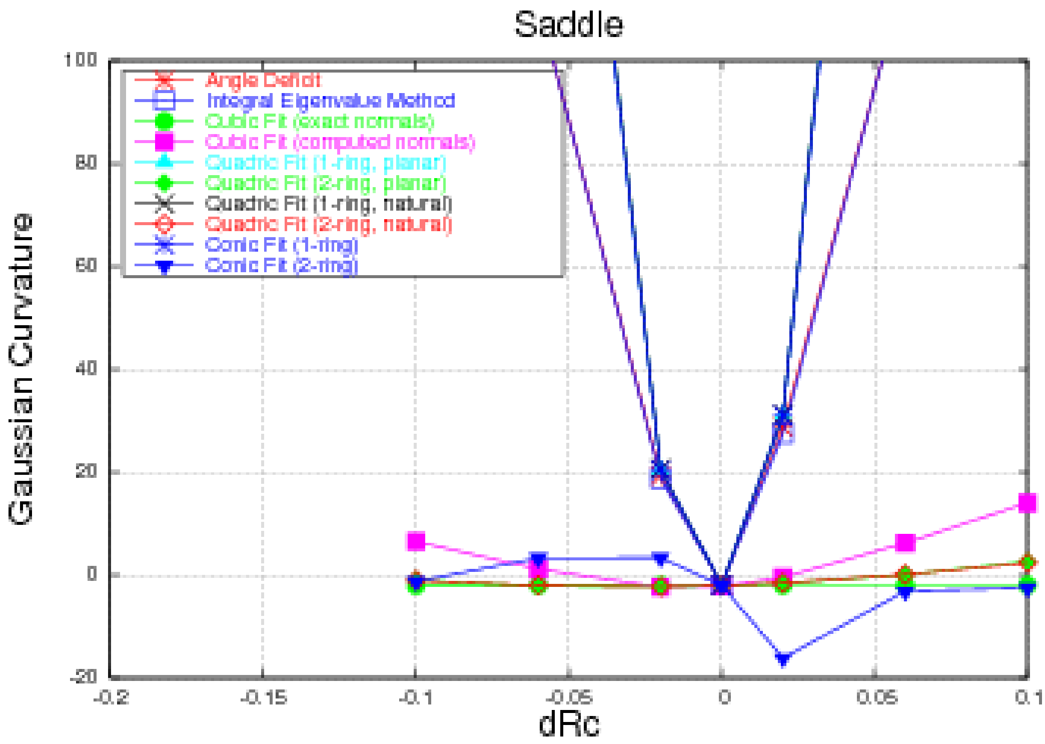

Fig. 4. Impact of noise normal to a saddle surface on the discrete and fitting methods. A valence of six was used for the data shown. The discrete methods and one-ring fitting methods exhibit extreme sensitivity to noise. The cubic fit behaves as a two-ring method and, along with the two-ring quadric fitting methods, shows the least sensitivity.

predicts the magnitude for both positive and negative Gaussian curvature.

Like the one-ring fitting methods, both of the discrete curvature estimation methods suffer from severe sensitivity to noise normal to the surface. But they are also very sensitive to perturbations of the mesh vertices along the surface, as shown in Figure 6. This is likely caused by the reliance on angles and areas of the mesh faces, which do not enter directly into the fitting methods.

3.4 Discussion of Results

Results for the mean curvature calculations were similar to those for Gaussian curvature. Mean curvature tends to be better behaved since it is an average rather than the product of the principal curvatures.

Figure 7 shows an example of the sensitivity of the principal direction estimate to mesh variation for the discrete methods and the cubic fit with computed normals. The principal direction calculations for the other methods were much less sensitive. In general, the principal curvature direction calcula-

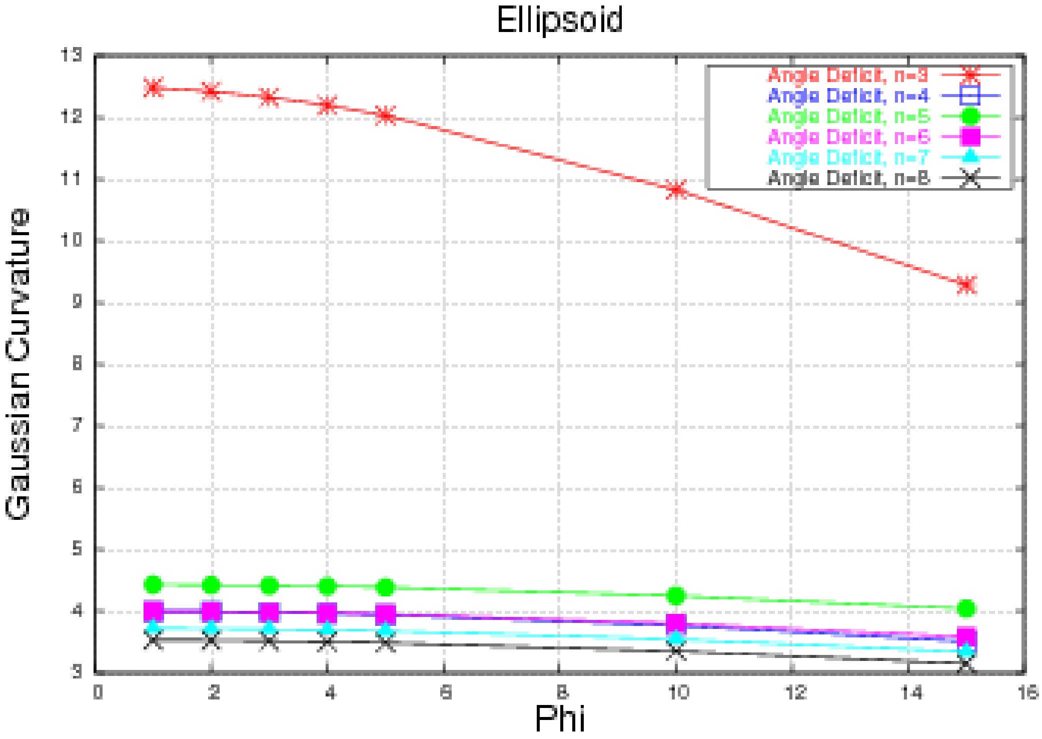

Fig. 5. Impact of valence on the accuracy of the angle deficit method, applied to an ellipsoid. Increasing ϕ represents decreasing mesh resolution. Only valence four and six converge to the actual Gaussian curvature (4.0). This method is extremely inaccurate for a valence of three, probably due to the effect of obtuse triangles.

tions appeared more stable than the curvature magnitudes. See Goldfeather and Interrante [2004] 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. And in fact the authors of these methods also have proposed associated smoothing algorithms.

Similarly, the one-ring fitting methods have higher noise sensitivity, and require smoothing if noise is present. 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. The cubic fit using surface normals is effectively a two ring method. This method performs very well if accurate normals are provided, but the effort required to generate accurate normals may negate its other benefits. It may still be the most appropriate method for calculating

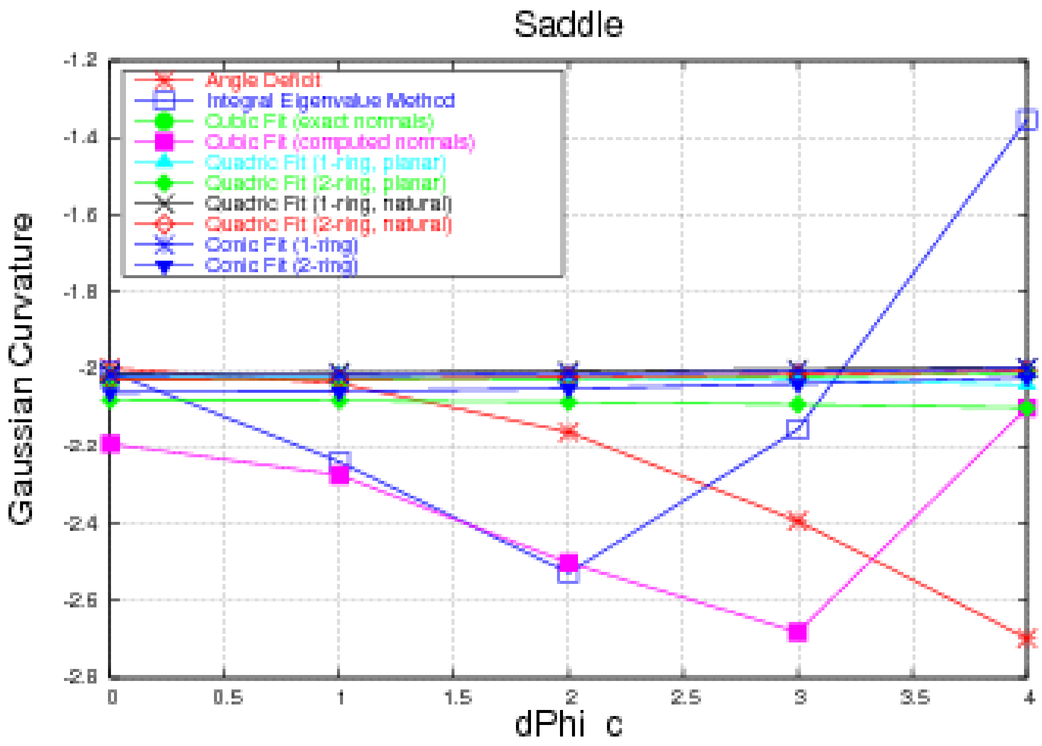

Fig. 6. Impact of moving the target vertex along the surface toward a one-ring point. The valence is six for all methods. The discrete methods and the cubic fit with computed normals are very susceptible to this mesh quality issue, while the other methods show little sensitivity.

curvature directions, as the stability of the directions seems independent of the accuracy of the curvature magnitudes.

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.

Fitting using radial basis functions did not yield suitable curvature estimates. However, there are a variety of possible formulations that may be worth investigating. Applying an alternate parameterization to the quadric fitting method showed more promise. The parameterization based on a flattening of the local mesh avoids potential problems due to folding or distortion when the mesh is projected to a plane. Behaviors for a two-ring fit based on a natural flattening [Desbrun et al., 2002] were similar to the two-ring planar fit, and in some cases accuracies were slightly better.

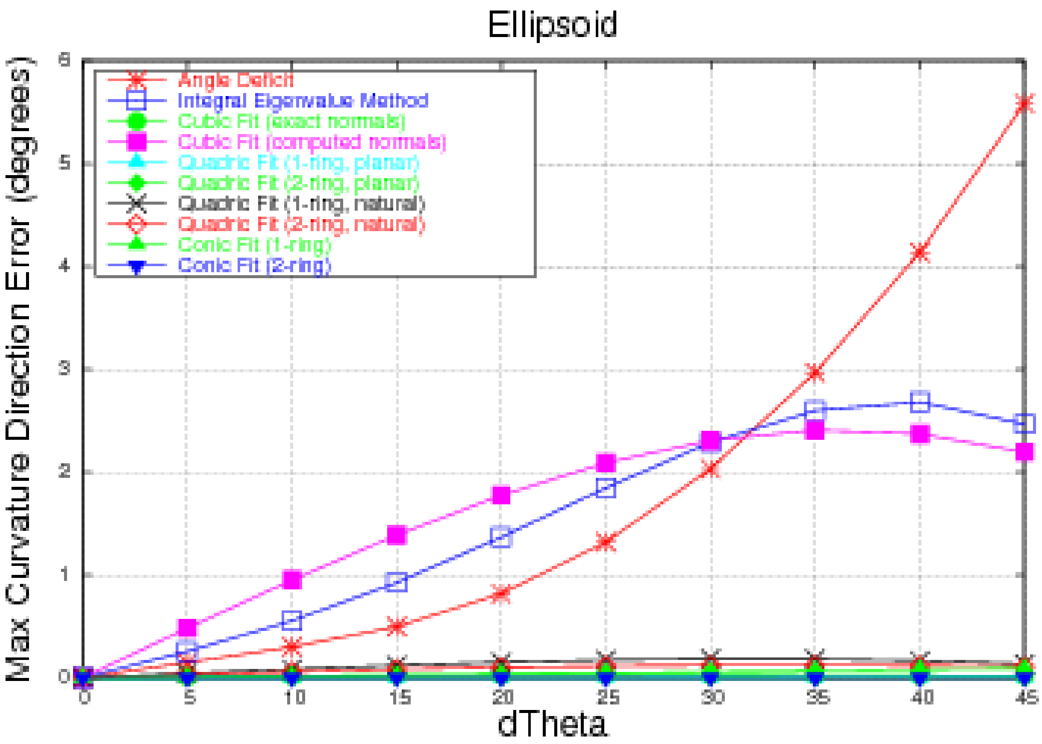

Fig. 7. Principal curvature direction error as a function of non-uniform spacing around the ring. A valence of six is used for the data shown. The discrete methods and the cubic fit with computed normals had extreme sensitivity relative to the other methods.

3.5 Extension to General Surfaces

These curvature estimation methods have been applied to a torus (with meshes having three different levels of resolution) and another general surface built from rational polynomials. Noise was simulated by randomly moving each vertex a random distance in a random direction. The curvatures were used to segment the surfaces based on the sign of the Gaussian curvature.

The mean and standard deviation of the error in the curvature estimates was also calculated. These statistics were compiled for the overall surface, and individually for two groups, vertices with positive Gaussian curvature, and those with negative Gaussian curvature. See Gatzke and Grimm [2003] for further details.

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. This suite has been applied to the most common surface fitting and discrete curvature estimation methods, to produce guidelines for choosing an algorithm.

The results also show that most methods have a bias toward over- or under-predicting curvature magnitudes. Further work will investigate if this bias, along with behavior based on mesh resolution and other factors can be used to place bounds on the error in the curvature estimates.

In comparing methods, it should be noted that some of these methods are based on formulas for the integral of curvature over an area, while others estimate curvature at a specific point. These methods will produce similar results if the curvature is relatively constant over the integration area, but may vary significantly in areas of rapidly changing curvature.

References

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.

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.

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.

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, 2000.

M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Eurographics 2002, Vol. 21, Number 2*, 2002.

I. Douros and B. Buxton. Three-dimensional surface curvature estimation using quadric surface patches. *Scanning 2002 Proceedings, Paris*, May 2002.

M. S. Floater. Parameterization and smooth approximation of surface triangulations. In *Computer Aided Geometric Design 14*, pages 231–250, 1997.

P. J. Flynn and A. K. Jain. On reliable curvature estimation. *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 110–116, June 1989.

T. D. Gatzke and C. M. Grimm. Assessing curvature metrics on triangular meshes. Technical report, Washington University, St. Louis Missouri, jun 2003.

Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/966131.966134.

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. ISBN 3-211-82399-9.

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.

A. Hilton, J. Illingworth, and T. Windeatt. Statistics of surface curvature estimates. *Pattern Recognition, Vol 28*, pages 1201–1221, February 1995.

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.

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. URL `citeseer.nj.nec.com/interrante97illustrating.html`.

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.

Alan McIvor and Robert J. Valkenburg. A comparison of local surface geometry estimation methods. *Machine Vision and Applications*, 10(1):17–26, 1997. URL `citeseer.nj.nec.com/ivor96comparison.html`.

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.

M. Meyer, H. Lee, A. H. Barr, and M. Desbrun. Generalyzed barycentric coordinates for irregular n-gons. In *Journal Of Graphic Tools*, 2002.

Peter T. Sander. Generic curvature features from 3-d images. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 1623–1635, November 1989.

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.

Alla Sheffer and Eric De Sturler. Smoothing an overlay grid to minimize linear distortion in texture mapping, 2002. URL `citeseer.nj.nec.com/sheffer02smoothing.html`.

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.

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.