

Washington University in St. Louis

## Washington University Open Scholarship

---

All Computer Science and Engineering  
Research

Computer Science and Engineering

---

Report Number: WUCSE-2004-9

2004-09-01

### Improved Curvature Estimation on Triangular Meshes

Tim Gatzke and Cindy Grimm

This paper takes a systematic look at calculating 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 three common surface fitting methods (polynomials, radial basis functions and conics). We also introduce a modification to the standard parameterization technique. Finally, we examine the behaviour of the curvature calculation techniques in the context of segmentation.

... [Read complete abstract on page 2.](#)

Follow this and additional works at: [https://openscholarship.wustl.edu/cse\\_research](https://openscholarship.wustl.edu/cse_research)

---

#### Recommended Citation

Gatzke, Tim and Grimm, Cindy, "Improved Curvature Estimation on Triangular Meshes" Report Number: WUCSE-2004-9 (2004). *All Computer Science and Engineering Research*.  
[https://openscholarship.wustl.edu/cse\\_research/1056](https://openscholarship.wustl.edu/cse_research/1056)

Department of Computer Science & Engineering - Washington University in St. Louis  
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

## Improved Curvature Estimation on Triangular Meshes

Tim Gatzke and Cindy Grimm

### Complete Abstract:

This paper takes a systematic look at calculating 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 three common surface fitting methods (polynomials, radial basis functions and conics). We also introduce a modification to the standard parameterization technique. Finally, we examine the behaviour of the curvature calculation techniques in the context of segmentation.



# Tech Report WUCSE-2004-9: Improved Curvature Estimation on Triangular Meshes

T. Gatzke<sup>1</sup> and C. Grimm<sup>1</sup>

<sup>1</sup> Department of Computer Science, Washington University, St. Louis, Missouri, U.S.A

---

## Abstract

*This paper takes a systematic look at calculating 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 three common surface fitting methods (polynomials, radial basis functions and conics). We also introduce a modification to the standard parameterization technique. Finally, we examine the behaviour of the curvature calculation techniques in the context of segmentation.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling, Curve, Surface, Solid, and Object Representations

---

## 1. Introduction

Many complex structures, ranging from natural objects such as bones to man-made ones such aircraft parts, are modeled as meshes. We would like methods to compare two meshes in order to, for example, identify similarities, quantify differences, or search for similar shapes in a database. One possible metric for use in surface comparison is *curvature*. Curvature is an intrinsic property of the surface and can be used to, for example, segment a surface into areas of positive and negative curvature. Unfortunately, curvature calculations on meshes tend to be very noisy<sup>10,9</sup>. One of the goals of this paper is to quantify, in a statistical sense, what kind of noise we might expect to see given that we know something about how the mesh was sampled. As we will show, each curvature technique responds differently to factors such as noise in the mesh, irregularities in the triangulation, and overall resolution.

Ultimately, we plan to use this information to create robust algorithms which incorporate *a priori* knowledge of the surface and sources of error in different curvature calculations. For example, if a curvature calculation technique is known to over-predict positive curvature, or fail in areas with discontinuities, then we can weight their results accordingly.

Meshes come from a variety of sources, such as scanning devices, analytical surfaces, or any number of other tech-

niques. The original analytical surface is rarely available, so several methods of computing the curvature directly on the mesh have been developed. They can be classified into two groups; discrete approximations based on the definition of curvature, or surface fitting approaches. We compare existing approaches (Desbrun's<sup>3</sup>, Taubin's<sup>15</sup>, Goldfeathers<sup>8</sup>, and polynomial surface fitting) plus several new fitting techniques we have developed (natural parameterization<sup>11</sup>, radial basis functions, and conics). These "new" techniques are standard approaches to surface fitting; we examine their suitability in the context of curvature calculation.

We would like to know how accurate these approximations are, and when they break down. Possible sources of error are the triangulation (*i.e.*, where on the surface samples are taken and how they are connected to form the triangulation), noise in the sampling process, and sampling density. We have developed 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 then evaluate the performance of these techniques in two test surfaces at different resolutions, the torus and a complex, analytical surface for which the curvature is known.

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 Gaussian curvature because it captures much of the data in a single number. The evaluation techniques easily extend to the other properties.

The three main contributions of this work are the development of several new curvature calculation techniques, the construction of a test suite for the evaluation of curvature calculation algorithms, and several insights into how and where different methods fail. We also discuss how these methods perform on a segmentation task.

Section 1.1 highlights previous work on curvature calculations for meshes. Section 2 briefly describes the different methods. In Section 3, the test cases, mesh parameters, and noise perturbations are described. Results of the analyses are presented in Section 4. Section 5 summarizes the conclusions of this study. Finally, Section 6 outlines possible areas for future work.

### 1.1. Previous work

A number of researchers<sup>6, 13, 2, 14, 1</sup> have looked at curvature estimation from 3-D range images for computer vision applications. While these methods were developed for 3-D range image data, the methods based on data fitting can also be applied to meshes. The main difference between curvature calculation on meshes and curvature calculations based on range data is that range data provides a rectangular array of pixel data, while on a mesh, data is available only at discrete points. Therefore the mesh-based approach requires a parameterization step before the fitting step. An obvious choice is to project the local portion of the mesh to a plane, which may cause folding. A better choice is to use a parameterization method such as Desbrun's<sup>11</sup> (see Section 11).

Meek and Walton<sup>12</sup> perform asymptotic analysis for several methods using both regular data (as in range data) and irregular data (as in meshes). However, they state that their asymptotic analysis applies only to discretization and interpolation methods, but not to least-squares fitting methods.

Desbrun et. al.<sup>4, 3</sup> defined methods to compute the first and second order differential attributes (normal vector, mean and Gaussian curvatures, principal curvatures and principal directions) for piecewise linear surfaces such as arbitrary triangle meshes. They claim optimality of their discrete curvature operators under mild smoothness conditions. They incorporate local operators to denoise arbitrary meshes of vector fields, while preserving features. However, they also admit that smoothing techniques do not deal well with large amounts of noise.

Taubin<sup>15</sup> proposed a method that estimates the tensor of curvature from the eigenvalues and eigenvectors of a  $3 \times 3$  matrix defined by integral formulas. He also incorporated a smoothing step for noisy meshes. A key benefit of his method is its simplicity; the complexity is linear in both time and space.

Goldfeather<sup>8</sup> compares several methods for calculating principal curvature directions, looking at the impact of random error on the resulting error in the calculated principal curvature directions. He also compares three methods for calculating normal directions at vertices. His primary test cases are a torus and a more complex closed surface, with random noise added to the vertices. He concludes that small errors in quantities such as normal curvature can amplify the error in the principal curvature directions. He describes an improved method that uses the normal vectors at adjacent points to generate a third-order fit in the curvature calculation. He attributes the improved control of the error magnitude for his cubic method to its third-order, rather than second-order, approximations.

## 2. Curvature Metrics

Curvature metrics can be grouped as either discrete metrics or fitting methods. In the following sections we describe both existing methods and our new or modified methods. The various methods and the acronyms summarized are summarized in Table 2.

| <i>Curvature Calculation Taxonomy</i>     |                  |             |
|---|------------------|-------------|
| Discrete Methods                          |                  |             |
| Discrete Curvature Operator               | DCO              |             |
| Modified DCO                              | Mod (NEW)        |             |
| Integral Eigenvalue Method                | IEM              |             |
| 1-Ring with Normal Fits                   |                  |             |
| Adjacent Normal Cubic<br>Exact normals    | ANCE             |             |
| Adjacent Normal Cubic<br>Computed normals | ANC              |             |
| Parametric Fitting                        |                  |             |
| Fit                                       | Parameterization |             |
|   | Planar           | Desbrun     |
| Polynomial                                | Fit*P            | Fit*N (NEW) |
| Radial Basis                              | RBF*P            | RBF*N (NEW) |
| Implicit Functions                        |                  |             |
| Conic Fit                                 | Con* (NEW)       |             |

\* = Number of rings

### 2.1. Discrete methods

Discrete methods attempt to calculate curvature directly from a given mesh without an intermediate fitting step. They derive relations based on a local region (typically the star of the vertex, or 1-ring) about a point.

### 2.1.1. Existing metrics

The discrete curvature operators of Desbrun et al.<sup>3</sup> compute curvatures for each vertex of a triangulation based on the 1-ring neighborhood around the vertex. An area (Voronoi area for non-obtuse triangles, and Barycentric area otherwise) is assigned to each of its vertices for use as a weighting factor in the algorithm. This area weighting causes problems for obtuse triangles, but can be fixed by changing the area weighting (see below and Figure 14). This method will be referred to as the Discrete Curvature Operators, or **DCO**.

The integral eigenvalue method of Taubin<sup>15</sup> estimates the tensor of curvature from the eigenvalues and eigenvectors of a  $3 \times 3$  matrix defined by integral formulas. This matrix is constructed as a summation around the 1-ring neighborhood using the projection of the edges to the tangent plane and an estimate of the curvature of that edge. Taubin recommends a smoothing preprocessing step for noisy surfaces. This method will be referred to as the Integral Eigenvalue Method, or **IEM**.

### 2.1.2. New metrics

The effect of obtuse triangles on the accuracy of the DCO method motivated our development of a variation of the DCO method with a slightly different area weighting formula. The modified method was the same as DCO for non-obtuse triangles, but the area weighting for obtuse triangles was formulated to avoid the discontinuity as a triangle transitions between non-obtuse and obtuse. In this formulation, the area of the triangle closest to a vertex is attributed to that vertex, which is consistent with the Voronoi area for a right triangle.

## 2.2. Fitting Methods

Fitting methods involve solving for an analytic function that approximates the mesh around a point, then calculating the curvature using that function. While any number of points can be used for the fit, the simplest approach is to fit an N-ring neighborhood around the point of interest. The N-ring neighborhood is an extension of the 1-ring neighborhood described above. A 2-ring neighborhood is created from the 1-ring neighborhood by adding all of the vertices of any face containing a vertex of the 1-ring neighborhood. Additional rings can be added in the same way. For this study we looked at 1, 2, and 3-ring neighborhoods.

Various functions can be used to fit to the points of the chosen neighborhood. The function can be either a parametric function (defined in terms of surface coordinates  $u, v$ ) or an implicit function (defined as the zero set of a function over  $x, y, z$ ). Each function has some number of free parameters which are solved for, usually using a least-squares approach.

Fitting methods require a parameterization. The simplest parameterization uses two orthogonal vectors in a local tangent plane perpendicular to the normal vector. This approach

can be sensitive to the selection of the normal vector and may cause folding. Alternatively, we can find a mapping that takes the vertices to the plane while minimizing some measure of distortion, such as the approach by Desbrun<sup>11</sup> (see Section 2.2.2).

### 2.2.1. Existing Metrics

A simple function that has been used in different forms by various researchers is a polynomial fit in parametric coordinates  $u$  and  $v$ . For a second-order polynomial with six coefficients in each coordinate direction we have:

$$x(u, v) = a_x u^2 + b_x uv + c_x v^2 + d_x u + e_x v + g_x$$

$$y(u, v) = a_y u^2 + b_y uv + c_y v^2 + d_y u + e_y v + g_y$$

$$z(u, v) = a_z u^2 + b_z uv + c_z v^2 + d_z u + e_z v + g_z$$

This method is equivalent to performing a coordinate transformation such that the positive  $z$  axis is aligned with the normal,  $x$  is aligned with  $u$  and  $y$  is aligned with  $v$ . In this latter form, referred to as a Monge patch,  $z$  is a height function of  $x$  and  $y$ . Both methods were implemented and the results agreed. This method will be referred to as a polynomial fit with planar parameterization.

Another common scattered-data surface fitting technique is radial basis functions. There are several possible choices for kernels; we chose a uniformly weighted Gaussian because the derivatives at a data point are well-behaved.

We also implemented the Adjacent-Normal Cubic Method of Goldfeather<sup>8</sup> described above. Goldfeather uses the normals at each vertex of the 1-ring neighborhood in a least-squares fit to generate a third-order surface approximation. Coefficients from this fit are used in the Weingarten curvature matrix. The eigenvalues and eigenvectors are then used to determine the principal curvatures and principal directions.

For the remainder of the paper, this method will be referred to as the Adjacent-Normal Cubic method, or **ANC**.

### 2.2.2. New Metrics

We implemented three new fitting methods. The first two are the polynomial fit and radial basis function fit described above, but with a different surface parameterization. The third technique involves fitting a general conic to the surface.

The simple parameterization based on a local tangent plane may cause folding or significant distortion of the spatial relationships between points, particularly near areas of high curvature. In general, it is not possible to map a 3-D mesh to a plane without introducing some distortion. However, several techniques exist<sup>11</sup> which produce a mapping

that minimizes some measure of distortion. A particular benefit of the method we chose is that the mapping of the boundary of the mesh to the plane is computed by solving for an optimal conformal mapping. Once  $u$  and  $v$  have been calculated for each vertex, they can be used to compute the polynomial or radial basis function fit just as was done for the tangent plane parameterization.

Conic surfaces, particularly ellipsoids, have been used for surface fitting, particularly in medical imaging applications. For completeness, we extend this approach to a general conic:

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

Douros and Buxton<sup>5</sup> apply a similar technique to a (typically dense) point cloud. In our initial implementation, the fit would occasionally use a very flat saddle-shaped conic such that adjacent points of the mesh would be on different branches of the conic. We found the fitting to be more stable if the points were transformed so that the normal was aligned with the  $z$  axis.

### 3. Noise Analysis Test Cases

In this section 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; the test cases are also suitable for evaluating other elements such as the maximum curvature.

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

The sphere equation:

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

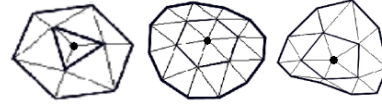
The ellipsoid equation:

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

The saddle equation:

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

The  $a$  and  $b$  parameters control both the maximum and minimum curvature at the point of interest. 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



**Figure 1:** Sample Test Case Meshes. Left: Sphere section ( $\phi=10$ , valence=3) Middle: Ellipsoid section ( $\phi=20$ , valence=6), Right: Saddle section ( $\phi=30$ , valence=5)

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 cases: 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. 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  $N > 1$  we can calculate approximate normal vectors at the target point and its  $N - 1$  ring neighbors.

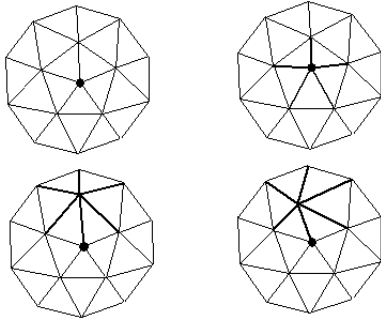
The Adjacent-Normal Cubic method uses normals to calculate the curvature at the target point. We calculate the curvature using both the approximate normals and the correct normals generated by the surface.

Several parameters control the qualities of the mesh. The number of points in the first ring is the first input parameter, with the second ring containing twice as many points. A second parameter,  $\phi$ , determines the distance from the target point to the first ring, and between the first and second 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 2-ring meshes are shown in Figure 1.

### 3.2. Noise Parameters

We define five components of noise, two of which are applied to the target vertex, and three of which are applied to the vertices in the first ring. The target vertex can be perturbed either in a direction normal to the surface, or along the surface (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 while maintaining a constant



**Figure 2:** Mesh Noise Components. Top left: Baseline, Top right: Target point moved on surface, Bottom left: Adjacent point moved away from center, Bottom right: Adjacent point moved along ring.

distance from the target vertex. Examples of perturbations along the surface are shown in Figure 2.

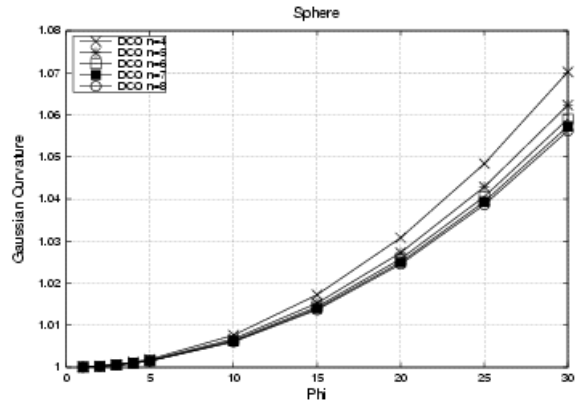
The perturbation component normal to the surface represents noise, *i.e.*, a true deviation from the actual surface geometry. The components along the surface, radial or circumferential, do not deviate from the true geometry, but rather represent the effects of mesh quality. The baseline 2-ring neighborhood 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 points, and are generally applied to one point at a time, to track the effects of the specific perturbation. As a result, the region around the target point is still, in general, better behaved than that of a general mesh point, which can have angle and edge length variations associated with each adjacent point.

### 3.3. Induced Random Noise

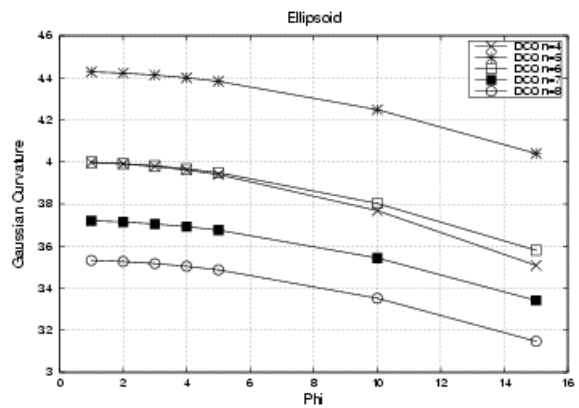
We can extend these noise-generation techniques to a complete mesh representing an analytic surface. The magnitude of the noise is specified based on a fraction of the smallest triangle edge around a vertex. We keep the magnitude of the noise below 50% to avoid (as much as possible) folding in the mesh.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

## 4. Results

The next several sections look at results for each of the individual curvature calculation methods. This is followed by comparisons between the methods, and then results of the impact of noise. Finally, application to general surfaces and segmentation of surfaces are presented.



**Figure 3:** Impact of mesh cell size ( $\phi$ ) for Discrete Curvature Operator on a spherical surface.



**Figure 4:** Impact of mesh cell size ( $\phi$ ) for Discrete Curvature Operator on an ellipsoidal surface.

### 4.1. Discrete Curvature Operator

Figure 3 shows a plot of the calculated Gaussian curvature on a sphere as a function of the mesh resolution parameter  $\phi$ . Separate curves are shown for valence (number of triangles sharing a vertex) four through eight. For these valences, the discrete curvature operator converges to the actual curvature value (1.0) for the sphere. As expected, the accuracy of the curvature estimate increases as the cell size decreases, representing a higher resolution mesh, or conversely, a smaller amount of curvature being modelled by each face. However, on a mesh with valence three, the DCO method has significant error. This can be attributed to the obtuse triangles in the mesh as supported by data from the mesh regularity variations discussed later. For coarse mesh resolution, there is a spread in the curvature accuracy with higher valence producing better curvature estimates.

Figure 4 indicates that the DCO method does not behave



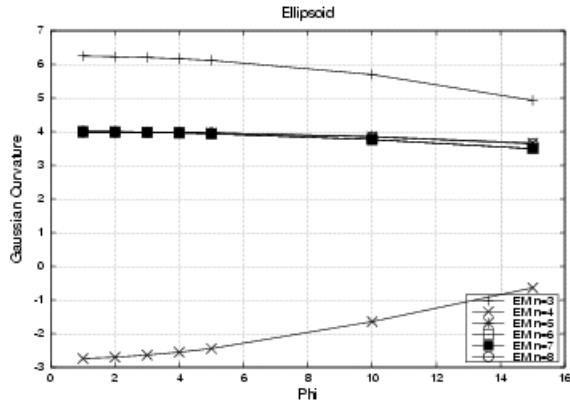


Figure 5: Impact of mesh cell size ( $\phi$ ) for Integral Eigenvalue Method on an ellipsoidal surface.

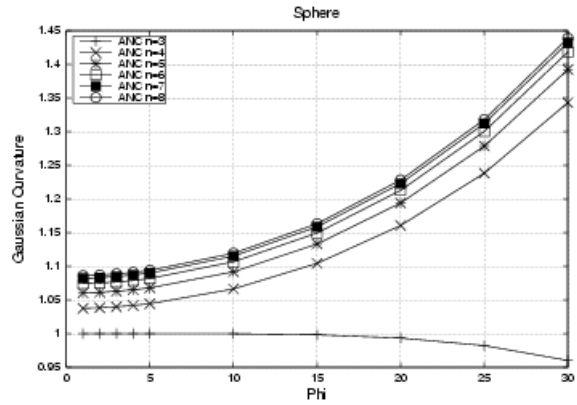


Figure 7: Impact of mesh cell size ( $\phi$ ) for Adjacent Normal Cubic (computed normals) on a spherical surface.

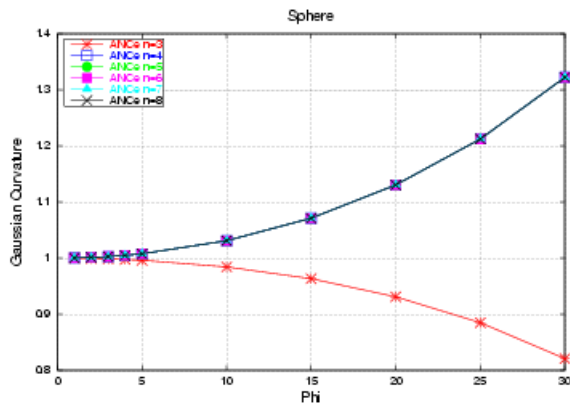


Figure 6: Impact of mesh cell size ( $\phi$ ) for Adjacent Normal Cubic (exact normals) on a spherical surface.

as nicely for the ellipsoid (similarly for the saddle-shape). The DCO method converges for valence 4 and 6 on the ellipsoid and for valence 6 on the saddle shape, but is off for other valences. These results indicate that the valence of the mesh vertices is important for the DCO method.

#### 4.2. Integral Eigenvalue Method

Figure 5 shows that the integral eigenvalue method is reasonably accurate for vertices where there are at least five triangles coming together, but performs poorly for valence three or four. The results for the saddle surface were similar.

#### 4.3. Adjacent-Normal Cubic Method

One of the main differences between the adjacent-normal cubic method and previous methods is that it uses more information, namely the surface normals at the adjacent vertices. For our test cases, we have an exact definition of the surface,

so we can compare the method based on the actual normals as well as normals calculated from the mesh.

Figure 6 illustrates the effect of cell size ( $\phi$ ) on the Gaussian curvature calculation using the adjacent-normal cubic method based on the exact normals for the spherical surface. Note the convergence from below for valence 3, while all the other valences converge from above, with little other effect of the valence. This also holds for the ellipsoid, but for the saddle shape, all valences converge from below, and there is more relative effect due to the valence for larger values of  $\phi$ , with larger valences being better than valence 3 or 4.

Figure 7 shows the same calculation for the adjacent-normal cubic method using normals computed from the mesh. This method does not appear to converge to the actual curvature, even for the sphere case. There is also significant variation due to the valence. Because these calculations do not include any noise, and are on the same mesh as the ANC method using exact normals, these errors must be attributed to errors in the calculation of the normals.

#### 4.4. Polynomial fit with tangent plane parameterization

Figure 8 shows how the 1-ring polynomial fit based on a planar parameterization converges for the saddle surface. All valences converge to the correct value, but there is more error for valence three. The 2-ring polynomial fit behaves similarly except that for the 2-ring fit valence 3 is more accurate than the other valences.

#### 4.5. Polynomial fit with Desbrun parameterization

Figure 9 shows the convergence for the 1-ring polynomial fit based on the Desbrun natural parameterization applied to the saddle surface. Note that the accuracy appears to be better for higher valences. This is probably due to having more

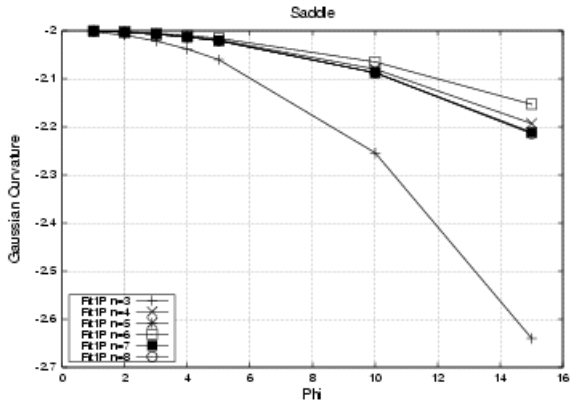


Figure 8: Impact of mesh cell size ( $\phi$ ) for a Polynomial (1-ring) fit based on planar parameterization on a saddle surface.

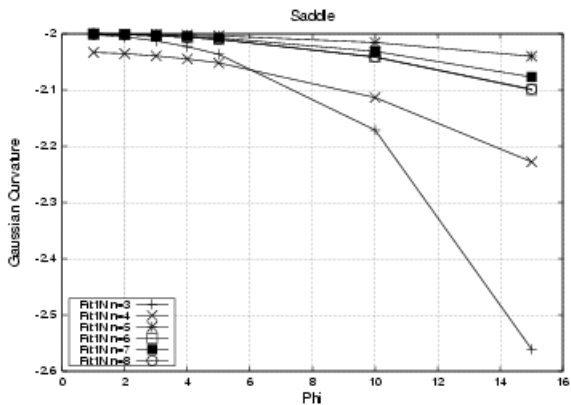


Figure 9: Impact of mesh cell size ( $\phi$ ) for a Polynomial (1-ring) fit based on a Desbrun parameterization on a saddle surface.

points used in the fit. The 2-ring and 3-ring versions of this method, with even more points, converge well.

4.6. Radial basis function fit

The radial basis functions converge for the planar parameterizations, but not the natural ones. They are very susceptible to noise in the normal direction, but not to mesh irregularity. Increasing the number of rings only slightly increases the accuracy. Note that most of the subsequent plots involving all metrics will not include this method because the error is an order of magnitude bigger in general.

4.7. Conic fit

The conic fit can model the sphere or ellipsoid exactly, producing the exact curvature for those cases, independent of

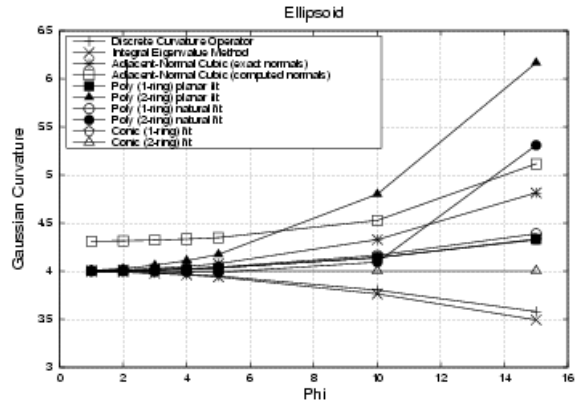


Figure 10: Comparison of curvature calculation methods on an ellipsoid.

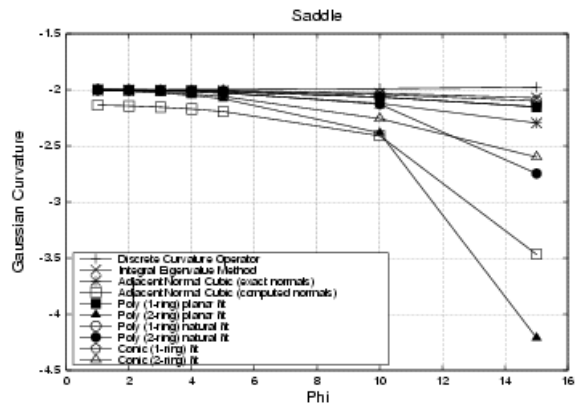


Figure 11: Comparison of curvature calculation methods on a saddle surface.

resolution. For the saddle surface, both the 2-ring and 3-ring conic fits showed essentially the same trends as the 2-ring polynomial fit based on the Desbrun natural parameterization.

4.8. Accuracy Comparison

Figures 10 and 11 highlight the relative accuracy of the methods in the absence of noise on an ellipsoid and a saddle surface, respectively. All methods are plotted for valence 6, which would be the preferred valence for an ideal triangular mesh. All methods except the ANC method using computed normals and the radial basis functions converge to the correct value. Note that while there are different rates of convergence, the trends in accuracy are not consistent between the ellipsoid and saddle shapes. The methods that have consistently better accuracy are the discrete curvature operators, integral eigenvalue method, polynomial fit using the Des-

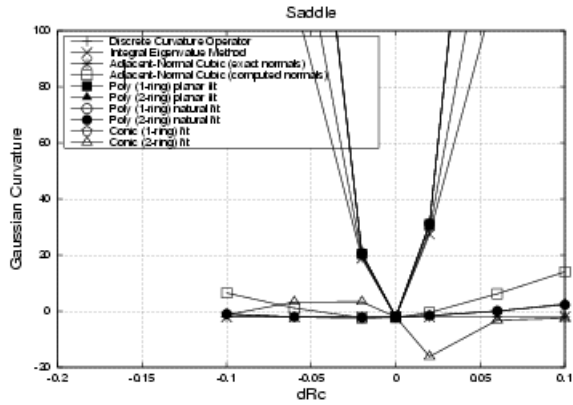


Figure 12: Impact of noise normal to the surface.

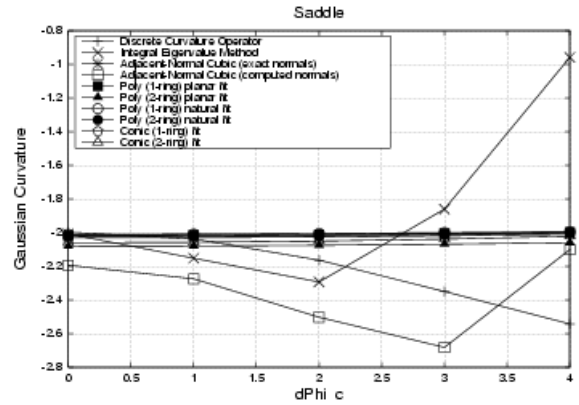


Figure 13: Impact of perturbations along surface normal to ring.

brun natural parameterization (1-ring and 2-ring), and the 1-ring polynomial fit using a planar parameterization.

#### 4.9. Noise Sensitivity

Figure 12 shows the impact of the noise component normal to the surface at the center point for a saddle surface. The discrete curvature operators, integral eigenvalue method, radial basis functions, and the 1-ring polynomial fits (planar and Desbrun parameterizations) were extremely sensitive to noise. The conic 2-ring fit and the adjacent normal cubic (computed normals) showed some noise sensitivity, while the adjacent normal cubic (exact normals), and the polynomial fits (planar and Desbrun parameterizations) using two or three rings showed the least sensitivity to noise normal to the surface.

Note that the methods which showed extreme sensitivity were the methods based on a 1-ring neighborhood. The methods based on data from two or more rings were much less sensitive.

Figure 13 shows the impact of the noise component along the surface, by moving the center point toward or away from a 1-ring point. Keep in mind that this noise really represents issues of mesh quality and not the integrity of the surface definition. The integral eigenvalue method, discrete curvature operators, and the adjacent normal cubic (computed normals) have significant sensitivity to this mesh characteristic, while the other methods show little dependence.

Figure 14 illustrates the effects of moving a 1-ring point along the ring toward one of its neighbors on an ellipsoid. Again, this represents issues of mesh quality and not the integrity of the surface definition. The discrete curvature operator shows some sensitivity up to  $\theta = 30.0$ , but for  $\theta > 30.0$  where obtuse triangles come into play, the accuracy degrades rapidly. The modified discrete curvature operator improves this behavior by making a smooth transition between the

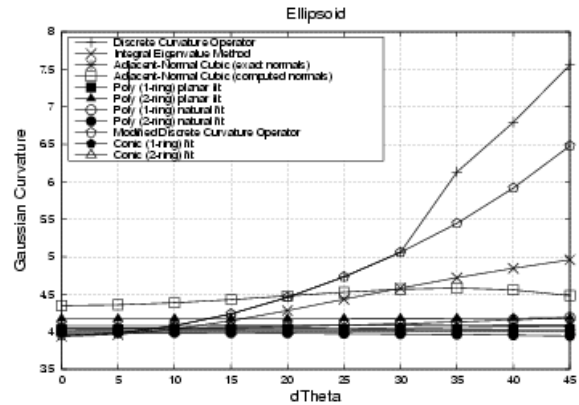


Figure 14: Impact of perturbations on an ellipsoid surface along ring.

area calculations as the angles become obtuse, however, the method is still the most sensitive to this mesh characteristic. The integral eigenvalue method also shows a sensitivity to this noise component. The remaining methods have very little dependence on this value.

#### 4.10. Extension to General Surfaces

How do the results above, from analyzing specific behavior for vertices with positive and negative Gaussian curvature, apply to general surfaces? We applied these methods to a torus (with meshes having three different levels of resolution) and another general surface built from rational polynomials.

##### 4.10.1. Curvature estimation on a torus

We applied the techniques to the baseline torus and to a torus with noise added. The noise was simulated by randomly

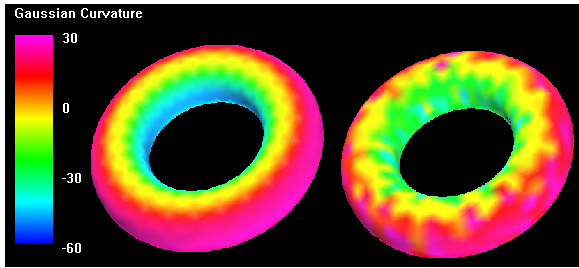


Figure 15: Surface plot of Gaussian curvature. Left: Exact. Right: Integral eigenvalue method.

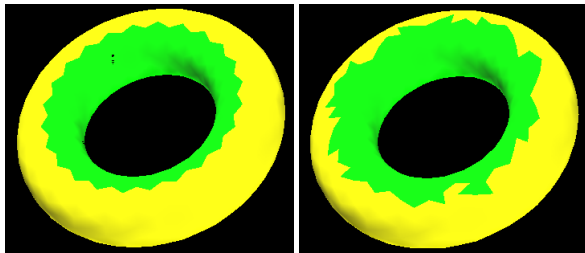


Figure 16: Segmentation. Left: Exact Gaussian curvature. Right: Integral eigenvalue method.

moving each vertex a random distance in a random direction. The maximum magnitude was 5.0% of the smallest spacing in the mesh. Figure 15 shows the surface of a torus colored by the exact Gaussian curvature value the integral eigenvalue method, plotted on the same scale. Without noise, most of the methods looked similar to the exact curvature plot, with slight variations in the magnitudes of the curvature values. There was more variation for the torus with noise, with the DCO and modified DCO methods looking similar to the integral eigenvalue method plots.

Next, we used these curvatures to segment the surfaces based on the sign of the Gaussian curvature. Figure 16 shows the segmentation using the exact curvature and the integral eigenvalue method. The DCO and modified DCO methods again looked similar to the integral eigenvalue method, while the other methods were more closely aligned with the exact segmentation.

We also calculated the mean and standard deviation of the error in the curvature estimates. These statistics were compiled for the overall surface, and into two groups, namely, the vertices with positive Gaussian curvature, and those with negative Gaussian curvature.

Figures 17, 18, 19, and 20 show logarithmic error plots for the 8x32, 16x48, and 32x96 resolution meshes for the torus for all methods. The error plots are capped at 120. For each method we split the error into two groups based on the sign of the *correct* curvature (negative on the left, positive on the right). If the error is positive, we draw a white bar. If

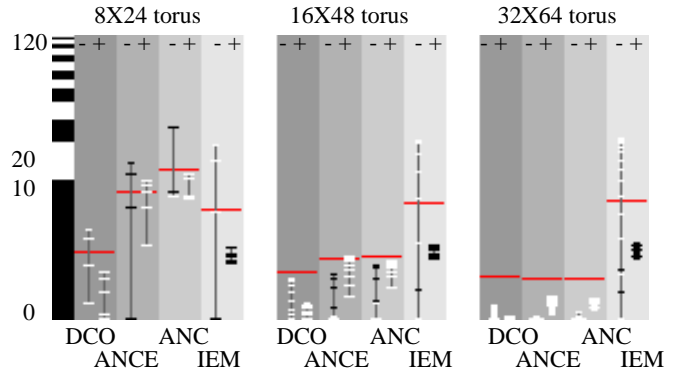


Figure 17: Logarithmic error on the torus for the four discrete methods.

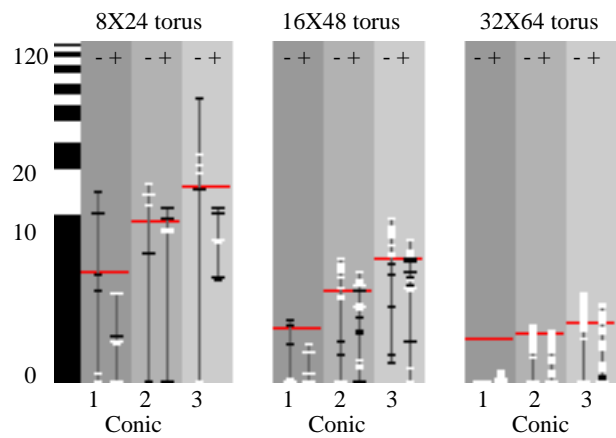


Figure 20: Logarithmic error on the torus for the conic methods (1-ring, 2-ring, 3-ring).

it is negative, we draw a black bar. The size of the bar indicates how many points were at this error level; smaller bars indicate  $< 0.1\%$ . The average is drawn in red across both the positive and negative errors. For example, in Figure 17, the DCO method has an average error of approximately 1.2 which decreases to less than one as the resolution of the torus is increased. The method also tends to over predict both negative and positive curvature.

All three meshes for the torus represent very regular triangulations. Although the torus has both positive and negative Gaussian curvature, the constant curvature in one direction and radial symmetry make this case fairly benign.

The DCO, ANC, ANCE, conic and polynomial methods all have very small error ( $< 1.0$ ) on the high-resolution mesh. The ANC method using computed normals is only slightly worse than the ANC method using computed normals, probably due to the regularity of the mesh. The IEV

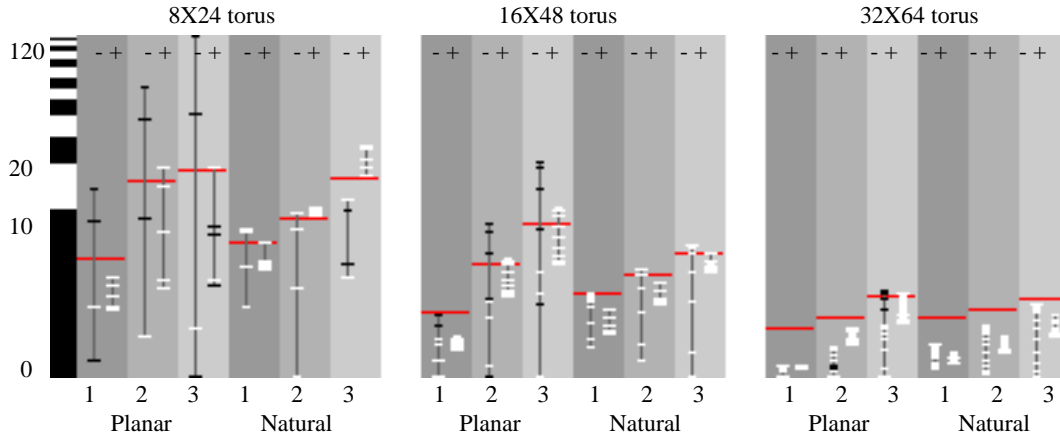


Figure 18: Logarithmic error on the torus for the four polynomial methods (1-ring, 2-ring, 3-ring, planar parameterization and Desbrun’s natural parameterization).

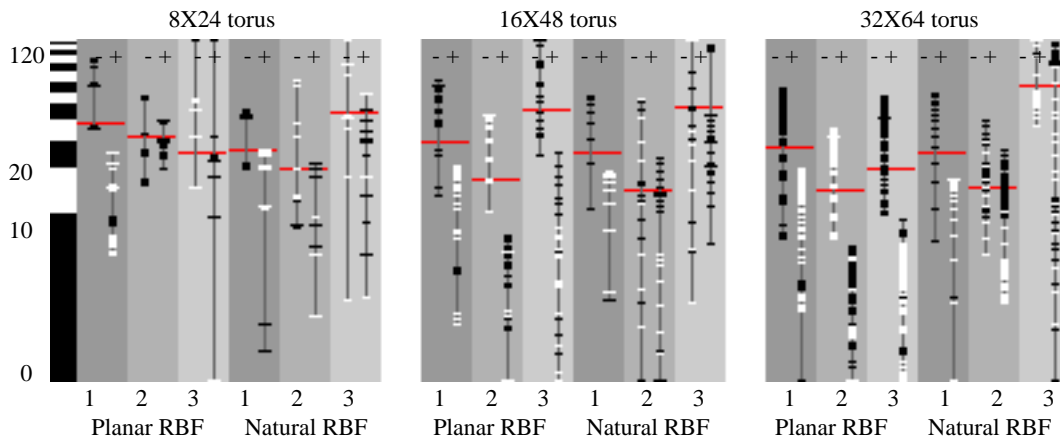


Figure 19: Logarithmic error on the torus for the radial basis function methods (1-ring, 2-ring, 3-ring, planar parameterization and Desbrun’s natural parameterization).

method shows no improvement as the mesh resolution is increased.

For all methods, the 3-ring planar parameterization on the low-res torus results in the mesh folding over on itself. This results in excessively large curvature values ( $> 1000$ ) for the polynomial and radial basis function methods for some points. The natural parameterization for the 2- and 3-ring polynomial fit prevented this folding, resulting in better accuracy. The radial basis functions saw a similar improvement for the 1- and 2-ring cases.

#### 4.10.2. Curvature estimation on a general surface

Figure 21 shows a more general surface built from rational polynomials <sup>7</sup>. The Gaussian curvature ranges from -25 to 63. The mesh tessellation is very regular (squares split into triangles).

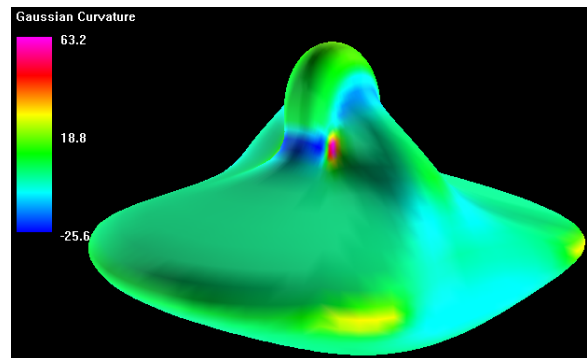
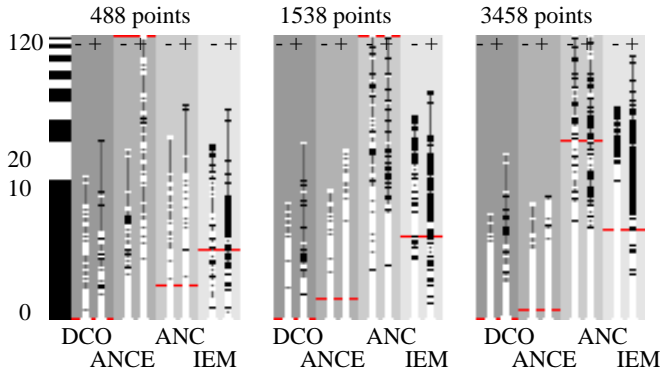


Figure 21: Surface plot of exact Gaussian curvature for a general surface with complex curvature.



**Figure 22:** Logarithmic error on the general surface for the four discrete methods.

The DCO, polynomial, radial basis functions, IEV, and conic methods exhibit the same relative behavior as they did on the torus, although the maximum error is higher. The ANC and ANCE methods, however, exhibited excessive curvature ( $> 1000$ ) for a small number of points (see Figure 22). These points all had the property that at least one of the edge normals was nearly orthogonal to the center normal, forcing the surface to “bow out”.

## 5. Conclusion

First, it is important to recognize that looking at surfaces colored by the calculated curvature values is not very useful for comparing methods. Noise, shape details of the surface, and the surface triangulation effect the accuracy of the curvature estimate and these effects are hard to detect by curvature visualization.

The overall mean and standard deviation of the error in the curvature estimates can also be misleading. For example, the ANC method has a small mean error for the torus, but this is due to a somewhat larger over estimation of positive curvatures and over estimation of negative curvatures that cancel each other out. By looking at the effects at vertices with positive and negative curvature separately, we can see trends for the torus such as, the ANC method over predicts both positive and negative curvature, the IEM method under predicts both positive and negative curvature, and the DCO and polynomial fit with natural parameterization over predicts positive curvature and under predicts negative curvature. Several of the other methods tend to have more error for negative curvature vertices than for positive curvature vertices. We found segmentation based on the sign of the Gaussian curvature less sensitive to error since, even with errors in the curvature magnitudes, the location where curvature is zero may not change if the errors are “balanced”.

The impact of noise in the surface data is significant for all of the 1-ring methods. Mesh irregularity can de-

grade the accuracy, particularly for the discrete methods and the adjacent-normal cubic with computed normals. Several methods are affected by the valence of the mesh. A valence of six, which is the “ideal” for a triangular mesh, was consistently good, but it is unrealistic to maintain exclusively for real meshes.

Our evaluations included several new methods. The modified discrete curvature operator showed a slight improvement in the behavior for obtuse angles, but the method is still sensitive to mesh regularity and noise. The conic fit methods performed fairly well on the torus, but did slightly less well than the polynomial method for the more complex surface. The radial basis function fit with the natural parameterization did poorly even on the torus, even though the radial basis functions were adequate using a planar parameterization. The polynomial fit with the natural parameterization showed promise, and was consistent for the torus and more complex surface.

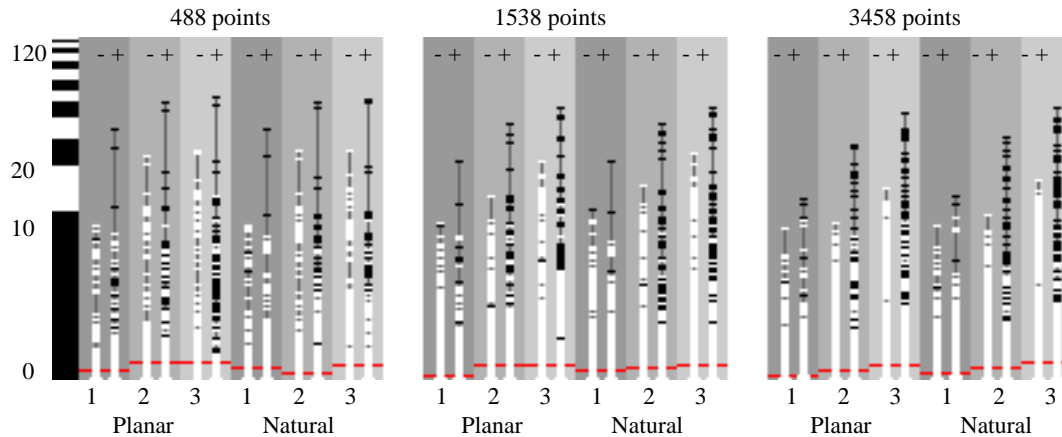
The radial basis function, polynomial, and adjacent normal cubic methods all have cases where the calculated curvature explodes. This happens because of folding, symmetry, and numerical instability for the radial basis functions, folding for the polynomial case, and orthogonal normals for the adjacent normal case. Both folding and orthogonal normals are detectable, so it might be possible to screen for these cases.

Different methods may be best for different circumstances. If it is known a priori that there is no noise in the data and that the mesh triangulation is well-behaved, the discrete curvature operator is a logical choice. In the presence of noise, the adjacent-normal cubic method is more robust, particularly if the normals can be computed accurately. The 2-ring polynomial fit with a natural parameterization is comparable in accuracy, very consistent, and very robust, although somewhat more expensive to compute.

## 6. Improvement Recommendations and Future Work

Opportunities exist to improve both the discrete curvature operators (better allocation of the area for obtuse triangles) and the adjacent-normal cubic method (improved normal calculation). Investigation of the effect of the valence is also needed. We would like to evaluate the discrete curvature operator combined with mesh smoothing<sup>16</sup> compared to using a metric such as the 2-ring polynomial fit that is less sensitive to noise. We would also like to investigate the effect of varying the weighting parameter in the Desbrun natural parameterization. In addition, these algorithms need to be applied to a wider variety of meshes to verify that the trends are consistent.

Our ultimate goal is to use the error bounds on the curvature and the parameterization techniques to develop more robust algorithms for identifying features based on curva-



**Figure 23:** Logarithmic error on the general surface for the four polynomial methods (1-ring, 2-ring, 3-ring, planar parameterization and Desbrun's natural parameterization).

ture and for use in quantifying the difference between objects represented by surface meshes.

#### Acknowledgements

The authors would like to thank Dr. Jack Goldfeather and Dr. Victoria Interrante for sharing mesh and curvature data for their complex surface test case, and insights and sample code for curvature calculations.

#### References

1. J. I. A. Hilton and T. Windeatt. Statistics of surface curvature estimates. *Pattern Recognition, Vol 28*, pages 1201–1221, Feb. 1995.
2. N. N. Abdelmalek. Algebraic error analysis for surface curvatures and segmentation of 3-d range images. *Pattern Recognition, Vol. 23*, pages 807–817, Sept. 1989.
3. M. Desbrun, M. Meyer, P. Schroder, and A. Barr. Discrete differential-geometry operators in nd, 2000.
4. M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics, 33(Annual Conference Series):317–324*, 1999.
5. I. Douros and B. Buxton. Three-dimensional surface curvature estimation using quadric surface patches. *Scanning 2002 Proceedings, Paris*, May 2002.
6. P. J. Flynn and A. K. Jain. On reliable curvature estimation. *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 110–116, June 1989.
7. T. D. Gatzke and C. M. Grimm. Assessing curvature metrics on triangular meshes. Technical report, Washington University, St. Louis Missouri, jun 2003.
8. J. Goldfeather. Understanding errors in approximating principal direction vectors. *preprint*, aug 2001.
9. A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 517–526. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
10. V. Interrante. Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution, 1997.
11. M. M. M. Desbrun and P. Alliez. Intrinsic parameterizations of surface meshes. *Eurographics 2002, Vol. 21, Number 2*, 2002.
12. 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, Feb. 2000.
13. P. T. Sander. Generic curvature features from 3-d images. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 1623–1635, Nov. 1989.
14. 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, Aug. 1992.
15. G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. pages 902–907, 1995.
16. G. Taubin. Geometric signal processing on polygonal meshes. 2000.