A Novel Cubic-Order Algorithm for Approximating Principal Direction Vectors

Jack Goldfeather and Victoria Interrante

Abstract

There are a number of applications in computer graphics that require as a first step the accurate estimation of principal direction vectors at arbitrary vertices on a triangulated surface. Although several methods for calculating principal directions over such models have been previously proposed, we have found in practice that all exhibit unexplained large errors in some cases. In this paper we describe our theoretical and experimental investigations into possible sources of errors in the approximation of principal direction vectors from triangular meshes, and suggest a new method for estimating principal directions that can yield better results under some circumstances.

Introduction

Suppose we are given only a surface mesh of vertices and polygons approximating some unknown smooth surface. There have been many methods proposed for approximating principal directions of the underlying surface [1,2,5,9,14,16]. In this paper we will examine a few of the known methods, showing how well they can work in some cases and how badly they can fail in others. In particular we will show how very tiny normal curvature approximation errors can be magnified into large errors in the estimated principal directions. We also introduce a new method that we believe performs significantly better under certain conditions than many other proposed methods. In section I, we briefly describe the motivation for this work and its significance to applications in computer graphics. In section II, we review the basic mathematics behind computing principal directions, stating the necessary formulas for the Weingarten curvature matrix and for its use in computing normal curvature in a given direction. In section III we describe in detail three methods each of which approximates the Weingarten curvature matrix at a vertex of the mesh. In section IV we apply each

method to a test surface using a number of different mesh schemes and then examine the direction errors. In the course of our investigations, we probe the relationship between surface approximation errors and errors in principal directions, and find that similar-appearing surface approximation errors can lead to vastly different principal direction errors. In particular, we find that while the chance for large errors increases near umbilical points, they can still occur at points on the surface where there is significant difference between the two principal curvatures. We conclude by summarizing the conditions under which principal direction estimation errors are most likely to occur, suggesting some steps that can be taken to improve the situation, and outlining promising directions for future work in this area.

I. Motivation

There are many applications for which it is useful to be able to calculate accurate estimates of the principal directions at points on a given surface. Principal direction vector fields have been widely used in shape analysis and surface interrogation in computer-aided manufacturing and design [8,12]. In these applications the surfaces are analytically defined, so the principal directions can be solved for directly and there are few problems with errors in the estimations. Principal direction vector fields have also been successfully used in conjunction with volumetrically-defined data to enhance the visual representation of surface shape for applications in molecular [4] and medical [11] visualization. In these applications the principal directions can be obtained by diagonalizing the Second Fundamental Form, whose entries can be fairly well approximated using first and second directional derivatives of the sampled 3D distribution.

Recently, there has been increasing interest expressed in the possibility of using principal direction vector fields over polygonal meshes for such purposes as guiding the direction of hatching strokes in pen-and-ink style renderings [6,10] or adaptively guiding the orientation of synthesized anisotropic texture patterns for enhanced surface shape representation [3,7]. However such efforts have been complicated by the lack of a robust and reliably accurate method for estimating the principal directions at points on an underlying smooth surface that is represented solely by a polygonal mesh. Although a number of methods for principal direction estimation have been previously published [16,9,2,5,1,14], what we have found in practice is that the computed vector field inevitably appears "noisy", requiring post hoc smoothing that can result in many

of the vectors falling out of alignment with the true principal directions and complicating our efforts to generate surface markings that closely follow the form.

In the research described in this paper we sought to gain insight into the sources of errors in principal direction estimation on surfaces defined by point samples organized in a polygonal mesh, in order to determine what the worst potential pitfalls are and how they might be overcome. Our goal was to enable an approach in which instead of accepting large principal direction estimation errors and then working to hide them, we could strive to obtain more accurate initial principal direction estimates while flagging the points at which the probability of obtaining erroneous estimates is high.

II. A Quick Review of Surface Curvature

Let p be a point on a smooth surface S, let N_p be the unit normal to S at p, and suppose $X(u,v)$ is a local parametrization of S in a neighborhood of p. Then using $X_u(p)$, $X_v(p)$, N_p as a local coordinate system, we can compute principal curvatures and principal directions as follows: Let λ_1 and λ_2 ($\lambda_1 \geq \lambda_2$) be the eigenvalues, and p_1, p_2 the associated unit eigenvectors of the Weingarten curvature matrix

$$W = \begin{pmatrix} \frac{eG - fF}{EG - F^2} & \frac{fE - eF}{EG - F^2} \\ \frac{fG - gF}{EG - F^2} & \frac{gE - fF}{EG - F^2} \end{pmatrix}$$

where

$$e = N_p \cdot X_{uu}(p) \qquad E = X_u(p) \cdot X_u(p)$$

$$f = N_p \cdot X_{uv}(p) \qquad F = X_u(p) \cdot X_v(p)$$

$$g = N_p \cdot X_{vv}(p) \qquad G = X_v(p) \cdot X_v(p)$$

Note that in the special case that X_u and X_v are orthogonal unit vectors, this becomes the symmetric matrix

$$W = \begin{pmatrix} e & f \\ f & g \end{pmatrix}.$$

If u is a unit vector in the tangent plane to S at p, then

$$\kappa_u = u^T W u$$

is the **normal curvature** of the surface in the direction of u.

It follows that λ_1 and λ_2 are the maximum and minimum normal curvatures of the surface at p, and $p_1 = \begin{pmatrix} p_{11} \\ p_{12} \end{pmatrix}$ and $p_2 = \begin{pmatrix} p_{21} \\ p_{22} \end{pmatrix}$ are the principal curvature vectors expressed in local coordinates. That is,

$$v_1 = p_{11} X_u + p_{12} X_v$$

$$v_2 = p_{21} X_u + p_{22} X_v$$

are the principal direction vectors in R^3.

An important observation is that at points where $\lambda_1 = \lambda_2$, the notion of principal direction is not defined, since all vectors are eigenvectors. Such a point is called an *umbilical* point on the surface.

III. Three Principal Direction Approximation Methods

The first step in computing principal directions on a surface mesh is to compute at each vertex p a vector N'_p that approximates the true unit surface normal N_p at p. Most methods compute a "normalized average" - i.e. a set of vectors is summed and the resulting vector is normalized to length 1.

Of course, using an incorrect surface normal will introduce error into a principal direction calculation. However, as we will show, even if the exact surface normal is used, there are *other* sources of significant potential error in principal direction estimation. In order to most effectively gain insight into these other sources of error, we decided to use exact surface normals in the subsequent steps of our initial investigations. Later, we return to examine the effects of using approximated surface normals.

The next step in estimating the principal directions is to choose a pair of orthonormal vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ in the plane through p with normal vector N'_p to form a local orthonormal coordinate system $L = \{\mathbf{x}_1, \mathbf{x}_2, N'_p\}$ in R^3. All subsequent calculations are done with respect to this local coordinate system. We examined three principal direction estimation methods, outlined below, each of which approximates the Weingarten curvature matrix expressed in this local coordinate system.

III.a The Normal Curvature Approximation Method

Suppose W is the unknown Weingarten matrix with respect to local coordinates L at vertex p. Suppose there are n vertices adjacent to p and let q_i denote the i^{th} adjacent vertex. We denote by $\mathbf{y}_i$ the unit vector obtained by projecting the vector $p\bar{q}_i$ (expressed in local coordinates L) onto the plane through p with normal vector N'_p and normalizing the result. Then using the result from section I, the normal curvature in the direction $\mathbf{y}_i$ is given by $\kappa_{\mathbf{y}_i} = \mathbf{y}_i^T W \mathbf{y}_i$. An approximation to this normal curvature is given by:

$$\kappa'_{\mathbf{y}_i} = 2\frac{(p-q_i)\cdot N'_p}{(p-q_i)\cdot(p-q_i)}$$

which is the curvature of the unique osculating circle passing through p and q_i with normal N'_p at p. This produces a system of equations:

$$\mathbf{y}_i^T W \mathbf{y}_i = \kappa'_{\mathbf{y}_i} \qquad i = 1, 2, \ldots, n \tag{1}$$

that we wish to solve for W. In [2], these equations are weighted in the same manner as the weighting in the third normal approximation method mentioned at the beginning of this section, but it is not clear to us that this improves the result much. The problem is that the osculating circle only produces a second-order approximation to the true normal curvature and second-order approximations can introduce significant error in many cases.

The first step in solving for W is to reorganize (1) as follows. Let $\mathbf{y}_i = (u_i, v_i)$ and

$$W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}.$$

Then

$$\mathbf{y}_i^T W \mathbf{y}_i = \begin{pmatrix} u_i & v_i \end{pmatrix} \begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} u_i^2 & 2u_iv_i & v_i^2 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix}.$$

If we let U be the $n \times 3$ matrix with rows $\begin{pmatrix} u_i^2 & 2u_iv_i & v_i^2 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} A & B & C \end{pmatrix}^T$, and d be the n-vector whose i^{th} entry is $\kappa'_{\mathbf{y}_i}$ the entire system can be written as the matrix equation

$$U\mathbf{x} = \mathbf{d}. \tag{2}$$

If $\mathbf{d}$ was the vector of true normal curvatures, this linear system would have an exact solution (assuming the adjacent vertices do not have some degenerate pattern, like all lying on the same line.) In practice, the

best we can hope to find is a least squares fit, i.e., a vector $\mathbf{x}$ that minimizes $||U\mathbf{x} - \mathbf{d}||$. There are standard numerical methods for finding this least squares solution. The resulting matrix

$$W' = \begin{pmatrix} A' & B' \\ B' & C' \end{pmatrix}$$

is used to approximate the principal directions. It is worth noting that the matrix U depends only on the projection of the adjacent vertices onto the local tangent plane, i.e. it is fixed by the choice of the mesh on the surface. The vector $\mathbf{d}$ depends on measurements we make at these adjacent vertices - i.e. it varies according to the amount of error we make. As we will see in a later section, it is possible to separate the mesh from the measurements so that we can understand how measurement error interacts with the local mesh geometry.

III.b The Quadratic Surface Approximation Method

In this method we try to best-fit a quadratic surface to the adjacent vertices. We begin by transforming each adjacent vertex q_i to local coordinates (x_i, y_i, z_i). In these local coordinates, p becomes $(0,0,0)$, N_p' lies along the positive z-axis, and the quadratic surface looks like

$$z = f(x,y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2.$$

It is easy to show that the Weingarten matrix for such a surface is

$$W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}.$$

As in the Normal Curvature method, we plug in the adjacent vertices to get a system of equations

$$\begin{pmatrix} \frac{1}{2}x_i^2 & x_i y_i & \frac{1}{2}y_i^2 \end{pmatrix} \mathbf{x} = z_i \qquad i = 1, \ldots, n.$$

As before, we can find a least-squares fit to this system. In order to be able to compare the result to the normal curvature method, we scale the i^{th} equation by $\dfrac{2}{k_i^2}$, where $k_i = \sqrt{x_i^2 + y_i^2}$. Then $(x_i, y_i) = k_i(u_i, v_i)$ so that

$$\begin{pmatrix} \frac{1}{2}x_i^2 & x_i y_i & \frac{1}{2}y_i^2 \end{pmatrix} = \frac{k_i^2}{2} \begin{pmatrix} u_i^2 & 2u_i v_i & v_i^2 \end{pmatrix}.$$

Then we can write each scaled equation in the system as:

$$\begin{pmatrix} u_i^2 & 2u_iv_i & v_i^2 \end{pmatrix} \mathbf{x} = d_i$$

where $d_i = \frac{2}{k_i^2} z_i$. There is a nice geometric interpreatation of d_i. Consider the parabola with equation $y = \frac{d_i}{2} x^2$. Plugging in $x = k_i$ we get

$$y = \frac{d_i}{2} k_i^2 = z_i$$

so that this parabola passes through the origin and the point (k_i, z_i). Since $k_i = \sqrt{x_i^2 + y_i^2}$, we can think of this parabola in three dimensions as passing through the local coordinate origin and the point (x_i, y_i, z_i).

The one variable, planar formula for computing curvature is :

$$\kappa = \frac{|y''|}{(1 + y'^2)^{3/2}}.$$

Using this formula for the parabola, at (0,0) we get $\kappa = |d_i|$. In other words, d_i is the normal curvature of a parabola passing through (x_i, y_i, z_i) and the origin. We can now see that the Quadratic Surface method is identical to the Normal Curvature method, except that curvature approximations are done using parabolas rather than circles. This suggests that these two methods will produce similar results, with one performing slightly better than the other depending on whether circles or parabolas better approximate the surface locally in some direction.

III.c The Adjacent-Normal Cubic Approximation Method

In both of the proceeding methods, we did not use all of the information available to us. Namely, we did not use the known (approximated) normal vectors at adjacent vertices q_i. We can use this information to create a third-order approximation method that we believe has not yet appeared in the literature. As we will see in the next section, this method seems to perform significantly better than the first two in many cases.

As in the Quadratic method we try to fit a surface to the adjacent vertex data. Let

$$f(x, y) = \frac{A}{2} x^2 + Bxy + \frac{C}{2} y^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3.$$

The Weingarten matrix for this surface is still $W = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$, because in the local coordinate system the curvature only depends on the second degree terms. However, using the third degree terms in the least-squares fit will produce values for (A, B, C) different from the ones we get in the quadratic method. The normal to this surface is given by:

$$N(x,y) = (f_x(x,y), f_y(x,y), -1)$$

$$= (Ax + By + 3Dx^2 + 2Exy + Fy^2, Bx + Cy + Ex^2 + 2Fxy + 3Gy^2, -1).$$

Let (a_i, b_i, c_i) denote the normal at the data point (x_i, y_i, z_i) (both normal and point must be transformed to the local coordinates), and let $\mathbf{x} = (A \quad B \quad C \quad D \quad E \quad F \quad G)^T$. Rewrite the normal as $(-\frac{a_i}{c_i}, -\frac{b_i}{c_i}, -1)$. Then for each point, we have an equation

$$(\,\tfrac{1}{2}x_i^2 \quad x_iy_i \quad \tfrac{1}{2}y_i^2 \quad x_i^3 \quad x_i^2y_i \quad x_iy_i^2 \quad y_i^3\,)\,\mathbf{x} = z_i$$

and for each normal we have two equations:

$$(\,x_i \quad y_i \quad 0 \quad 3x_i^2 \quad 2x_iy_i \quad y_i^2 \quad 0\,)\,\mathbf{x} = -\frac{a_i}{c_i}$$

$$(\,0 \quad x_i \quad y_i \quad 0 \quad x_i^2 \quad 2x_iy_i \quad 3y_i^2\,)\,\mathbf{x} = -\frac{b_i}{c_i}.$$

As in the preceding method, we scale these equations by the same scale factor $\frac{2}{k_i^2}$, to obtain a system

$$U\mathbf{x} = \mathbf{d}$$

where U is a $3n \times 7$ matrix and $\mathbf{d}$ is a $3n$-vector. Again, we find a least-squares fit, but use only A, B, and C from the result. Note that n has to be at least 3 for there to be at least as many equations as unknowns.

III.d Higher Order Methods

The preceding method can be extended to higher orders:

$$f_4(x,y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + \sum_{n=0}^{n=3} D_n x^{3-n}y^n + \sum_{n=0}^{n=4} E_n x^{4-n}y^n$$

$$f_5(x,y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + \sum_{n=0}^{n=3} D_n x^{3-n}y^n + \sum_{n=0}^{n=4} E_n x^{4-n}y^n + \sum_{n=0}^{n=5} F_n x^{5-n}y^n$$

and so on. However, in order that there be at least as many equations as unknowns, the maximum degree of the approximation that we can compute is limited by the number of adjacent vertices in the mesh. For example, if we use 6 adjacent vertices, then there will be 18 equations. The highest order method that we can use is then degree 5, since such an approximation will have the 18 unknowns $(A, B, C, D_0, \ldots, D_3, E_0, \ldots, E_4, F_0, \ldots, F_5)$.

As we will see in what follows, higher order methods seem to make small errors smaller, but can also make large errors larger.

IV. Testing the Methods

We set out to design a test surface that had significant areas where the curvature was "interesting". Our surface, $S(u,v)$, is defined by:

$$f(u) = -2u^4 + 2u^2 + u/6 + 0.3$$

$$g(u) = \begin{cases} .9 + u + f(-.9), & -.9 - f(-.9) \le u \le -.9; \\ f(u), & -.9 \le u \le 1; \\ 1 - u - f(1), & 1 \le u \le 1 + f(1); \end{cases}$$

$$h(u) = \begin{cases} -.9, & -.9 - f(-.9) \le u \le -.9; \\ u, & -.9 \le u \le 1; \\ 1, & 1 \le u \le 1 + f(1); \end{cases}$$

$$x(u,v) = g(u)\cos(v)$$

$$y(u,v) = g(u)\sin(v)$$

$$z(u,v) = h(u) + 0.2\sin(2x(u,v)) + 0.15\cos(3x(u,v)y(u,v))$$

$$S(u,v) = (x(u,v), y(u,v), z(u,v)) \qquad .9 - f(-.9) \le u \le 1 + f(1), \qquad 0 \le v \le 2\pi.$$

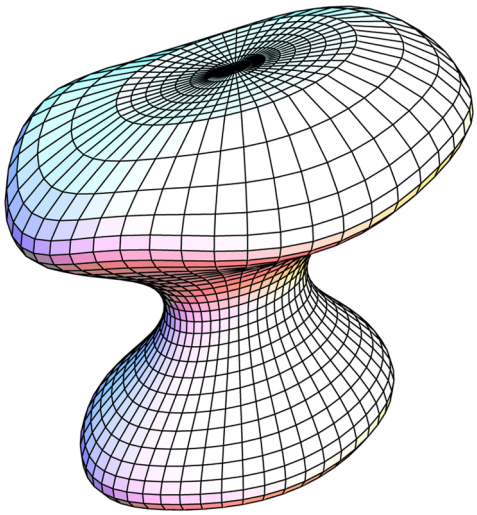

Figure 1. Test Surface

IV.a Global Parameter Space Triangulations

For our first set of tests we created two different global mesh representations of the surface. First, we created a Global Regular triangulation of the parameter space. This is the type of meshing condition that we had found most commonly used for demonstrating the results of previous principal direction estimation algorithms. Figure 2-left shows an example of a 4 by 4 global regular triangulation. The actual triangulation we used was 50 by 50.

Then we created a Global Random triangulation by jittering the locations of the interior points within the parameter space. This more closely resembles the type of meshing condition that we had encountered in practice, when desiring to estimate the principal directions at the vertices of arbitrary polygonally-defined models. Figure 2-right shows an example of a 4 by 4 global random triangulation in parameter space.

By definition, all of the object space vertices obtained from each of these parameter space triangulations lay exactly in the analytically defined surface.

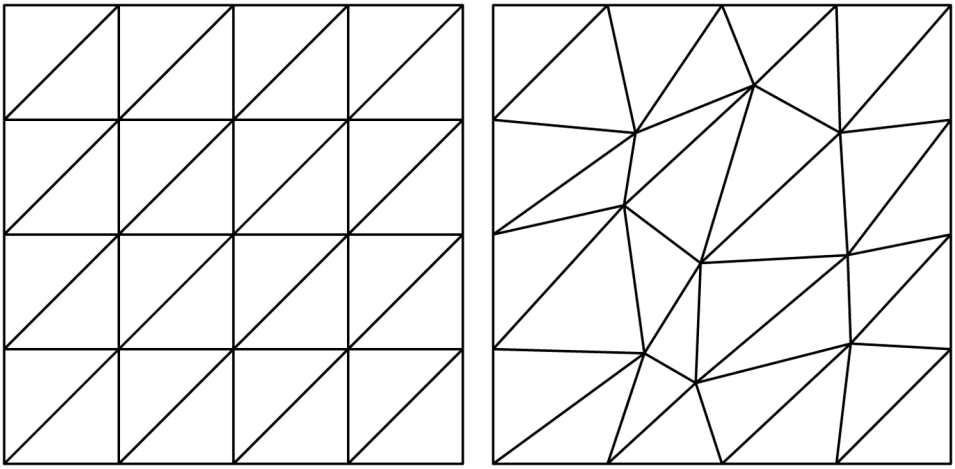

Figure 2. Left: Regular Triangulation in parameter space, Right: Random Triangulation in parameter space.

Figure 3 shows the results of estimating principal directions using each of the three principal direction approximation methods described in section III, in the case where the data is represented by the Global Regular mesh. All of the methods perform very well under this condition.

Figure 4 shows the estimated principal directions computed using the same three methods at the vertices of the Global Random mesh. In this case, we found a dramatic increase in the prevalence of large angle errors in the principal direction estimates, even at points that were far from being umbilic. The increase in errors is especially severe with the two second order methods. Estimated directions more than 10° out of alignment with the true principal directions are highlighted in red.

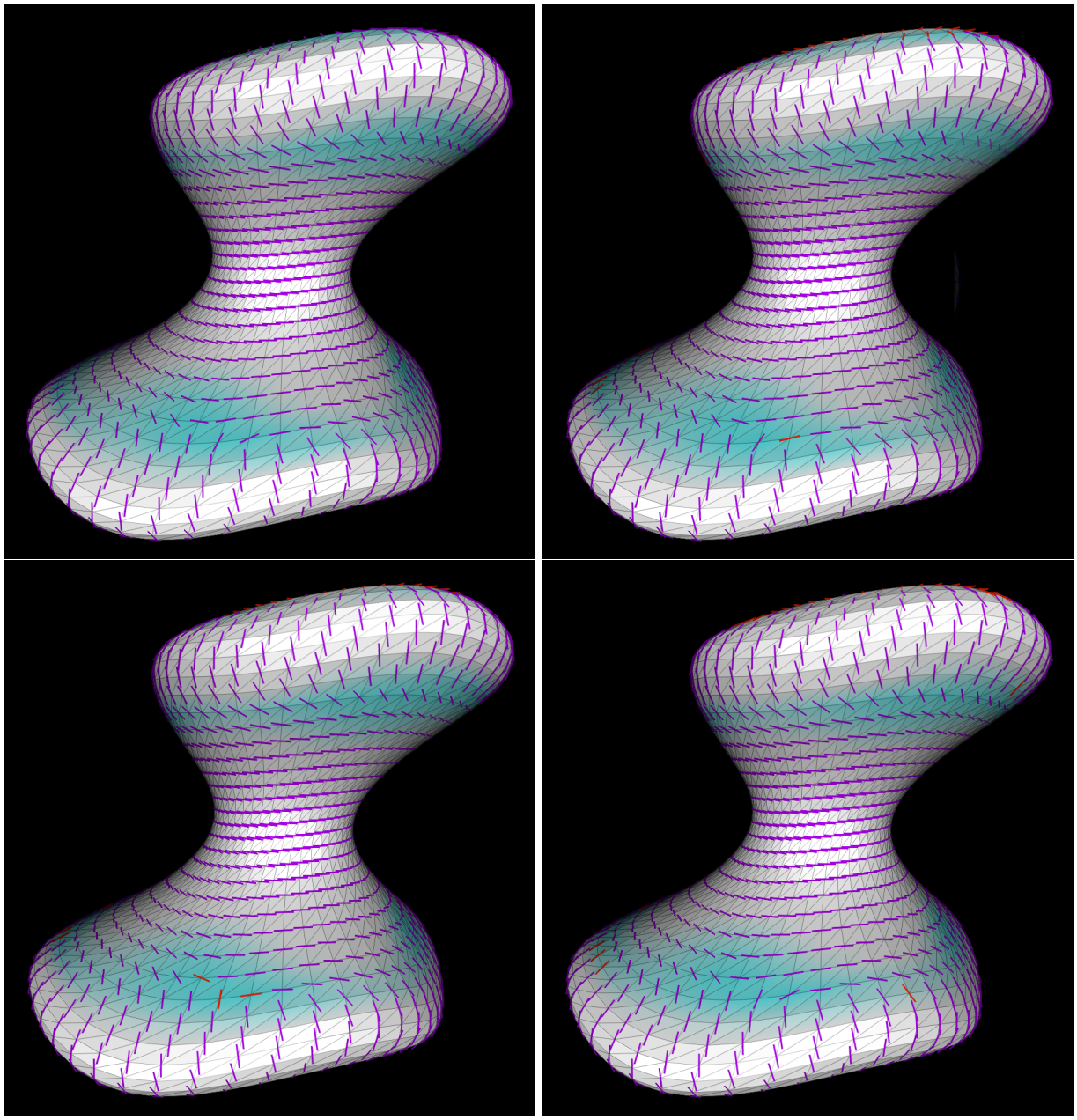

Figure 3. Approximated principal directions, Global Regular mesh. Directions marked in red have errors above 10°. Surface coloration highlights areas in which the difference between the two principal curvatures becomes small (near-umbilic points). Upper left: exact directions. Upper right: approximated directions, Quadratic Method. Lower left: approximated directions, Normal Curvature Method. Lower right: approximated directions, Cubic Method.

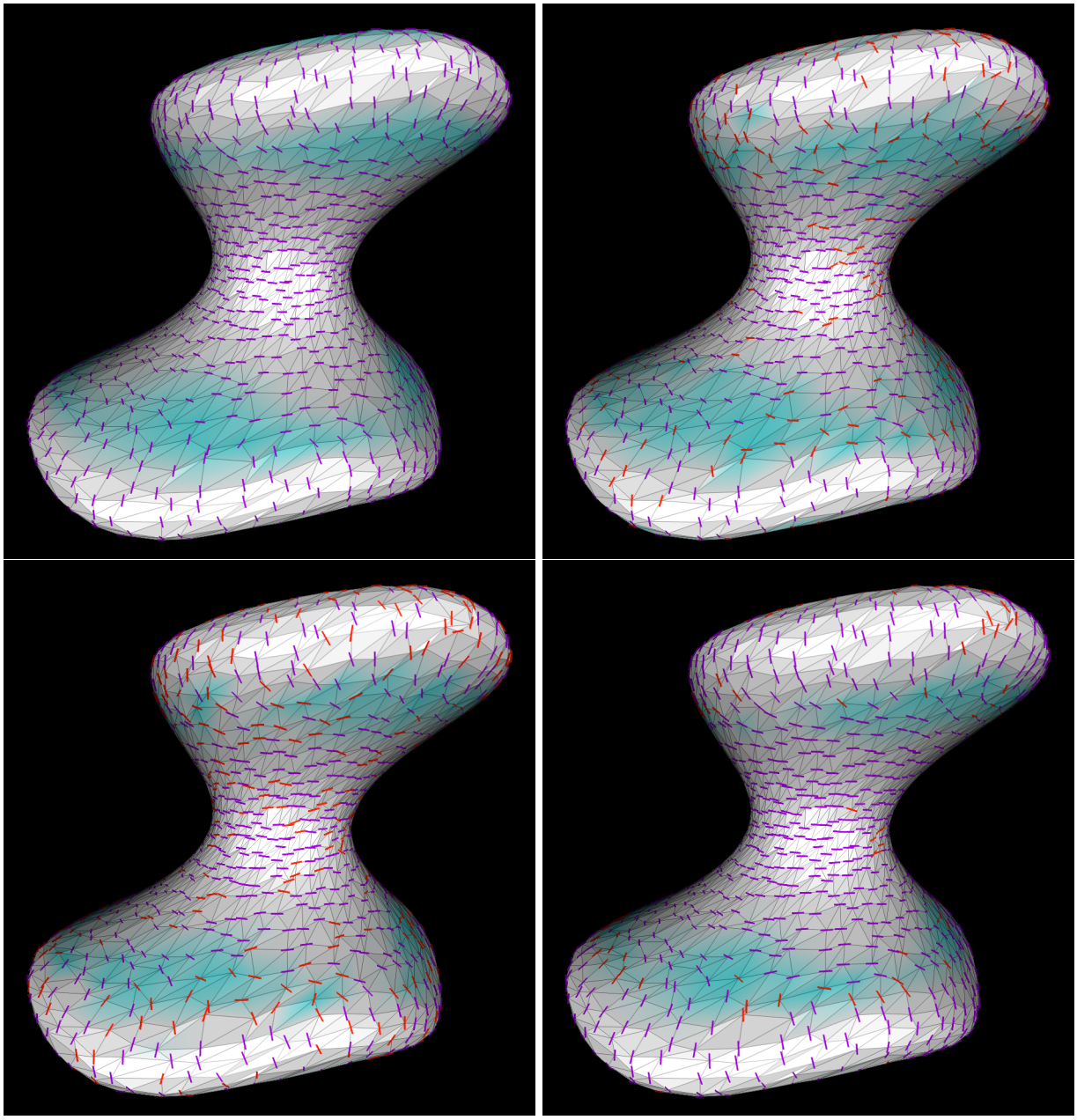

Figure 4. Approximated principal directions, Global Random mesh. Directions marked in red have errors above 10°. Surface coloration highlights areas in which the difference between the two principal curvatures becomes small (near-umbilic points). Upper left: exact directions. Upper right: approximated directions, Quadratic Method. Lower left: approximated directions, Normal Curvature Method. Lower right: approximated directions, Cubic Method.

IV.b Local Tangent Space Projections

Seeking deeper insight into the source of the large errors in principal direction estimates under non-regular meshing conditions, we set out to devise a number of different local triangular meshes on the surface that we could use to investigate the following questions:

1. Does regularity of the mesh improve accuracy and randomness reduce it?
2. Does enforcing equiangularity between adjacent vertices improve accuracy?
3. Does enforcing equidistance to adjacent vertices improve accuracy?
4. Does choosing adjacent vertices that lie along the true principal directions have any effect on accuracy?

With these in mind, we created six local meshing patterns, which we defined independently around each of 575 individual vertices randomly sampled from the global random mesh. None of the 575 local mesh center points were umbilical points. Roughly 80% satisfied $\lambda_1 - \lambda_2 > 0.5$; the remainder had $\lambda_1 - \lambda_2$ reasonably uniformly distributed between 0 and 0.5.

To obtain the local surface meshes, we began by defining, in parameter space, the six planar mesh patterns shown in Figure 5. We achieved each of these patterns by rotating a single radial edge about a given centerpoint according to various criteria. In all cases a total of six edges were defined, with the first edge aligned with the analytically-determined first principal direction. For the first local mesh, we used a constant angle of rotation (60°) to define all of the edges. With this mesh we would be able to test whether enforcing equiangularity between adjacent vertices improves the accuracy of the principal directions estimated at the central point. We refer to this mesh type as "local equi-angle". In the second case, we defined the angles between the edges to lie in a regular pattern with 3-way symmetry, by alternating 30° with 90° rotations. We chose this pattern in order to test the question of whether mesh 'regularity', in terms of a certain symmetry in the locations of edges, improves the accuracy of the results. We refer to this mesh type as "local 30-90". In the remaining four cases we defined the meshes by randomly choosing five successive angles of rotation in $[1^\circ \; .. \; 90^\circ]$. We refer to these meshes as "local random [1-4]". We felt that it was important to additionally test the methods on a reasonable set of randomly defined local meshes in order to establish a larger context within which to interpret the results we would find in the more highly

constrained conditions.

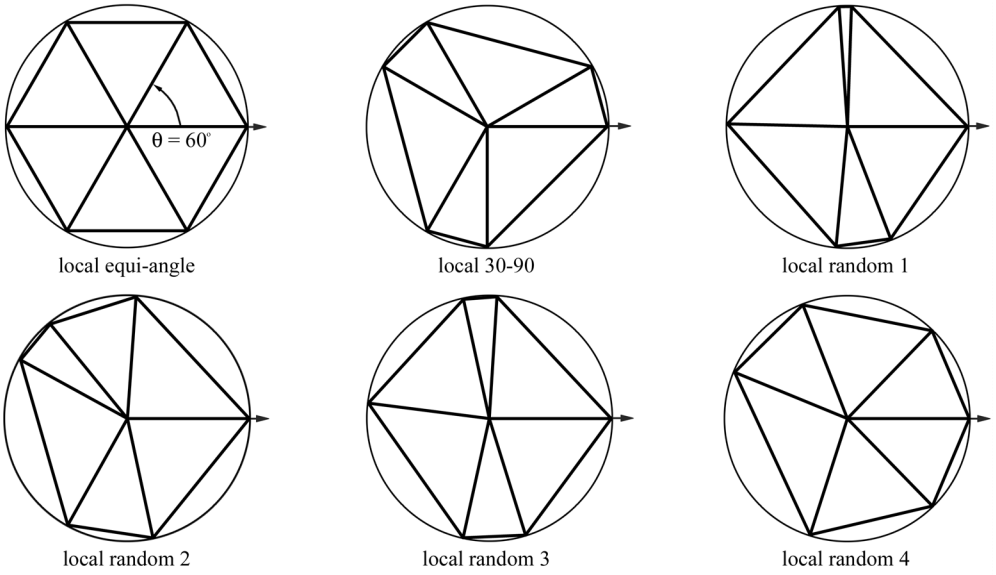

Figure 5. Our six local mesh patterns in parameter space.

For each of the six local mesh patterns in parameter space we obtained a collection of 575 local surface meshes by determining the points on our analytically-defined test surface that projected onto each pattern when the pattern was laid in the tangent plane to the surface at each of the 575 sample points (Figure 6).

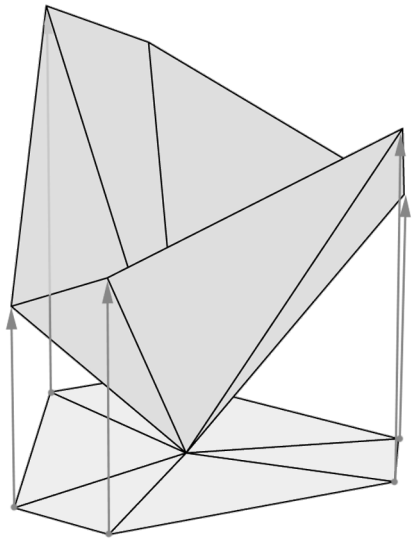

Figure 6. Diagram showing how a local surface mesh is obtained from a local mesh template in the tangent plane.

Finally, we rotated each of the patterns in the tangent plane 5 times (72 degrees each time). We did this in order to investigate the hypothesis that having some mesh edges coincidentally aligned with or counter-balanced against the principal directions might affect the accuracy of the approximation. The initial orientation of each local mesh was configured to have one edge lying along a principal direction. Subsequent rotations moved this edge away from the principal direction.

IV.c Findings and Interpretation

The results of our further testing are summarized in Figures 7-8. Detailed tables are available on the website.

Figure 7 shows the median angle error, in degrees, obtained with each of the three principal direction estimation methods, applied both to the two global meshes and to each of the six different local meshes at 575 sampled points on the surface. Median errors for the local mesh results were computed using data from all five mesh rotations, or 2875 values in all.

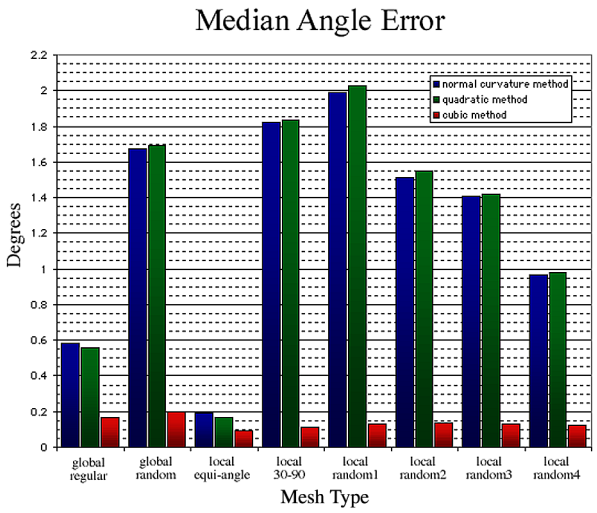

Figure 7. Median angle error in the approximated principal directions, per mesh type and method.

Overall, performance was best when the Cubic method was used, and was not as good in the cases of the Quadratic and Normal Curvature methods. The performance in the cases of the two second order methods was very similar. The Cubic method exhibited the least variation in performance over the different

mesh conditions. With the two second order methods, the size and prevalence of errors varied significantly according to the mesh type, indicating that some meshing arrangements were 'better' than others. The smallest errors occured in the local equi-angle and global regular cases, and the largest errors occured in the cases of the global and local random meshes and the local 30°-90° pattern.

Figure 8 provides a more detailed look at the data for the normal curvature and cubic methods, showing the percentage of mesh points at which the angle errors of the estimated principal directions fell below specific thresholds.

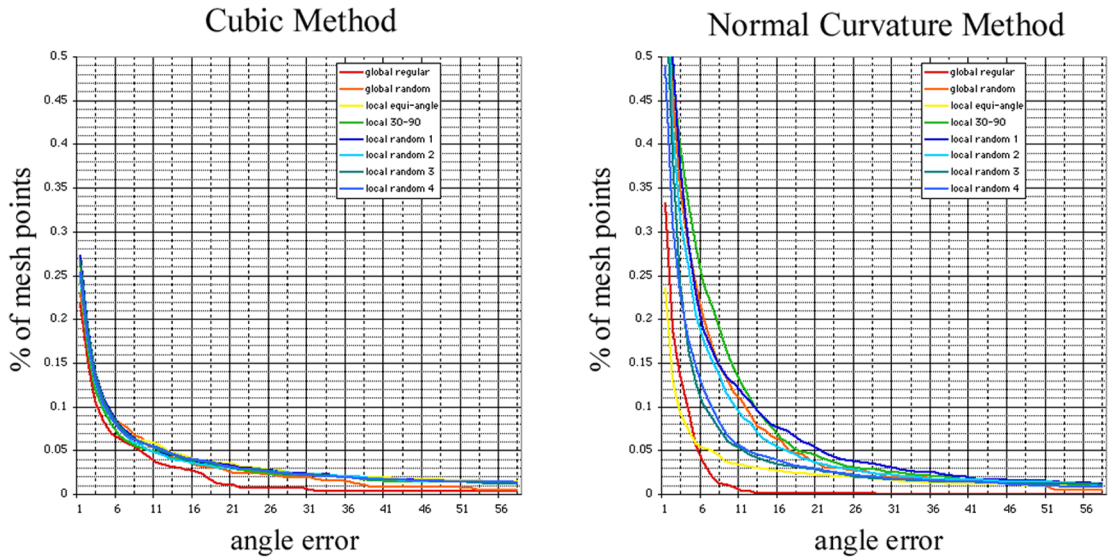

Figure 8. Prevalence of errors in the approximated principal directions, by magnitude.

In figure 8a we can again see that with the Cubic method, performance was fairly consistent across all mesh types. Approximately 22-27% of the mesh points had angle errors above 1°, with that number rapidly dropping to 5% with errors above 10°-15° and below 3% with errors above 20°, for all mesh types. In figure 8b we can see again that in the case of the second order method, there is greater variation in performace under the different mesh conditions. In aggregate, between 23% to over 50% of the mesh points had angle errors above 1°, with that number dropping to 5% at thresholds between 5°-22°, depending on mesh type. For this method, performance was particulary good in the cases of the global regular mesh, where fewer than 1% of mesh points had errors above 10°, and the local equi-angle mesh, where 90% of mesh points had errors below about 3°, but poorer in the other mesh conditions.

Table 2, provided on the website, contains numeric data that lists, for each meshing scheme and each local mesh rotation, the number of points at which the degree error falls into each of three ranges: less than 3°, between 3°-6°, and more than 6°. Figure 9, derived from Table 2, illustrates the range of variation in angle errors found under the different mesh rotation conditions, for the cubic and normal curvature methods. We noticed no apparent advantage for the principal-direction aligned rotations.

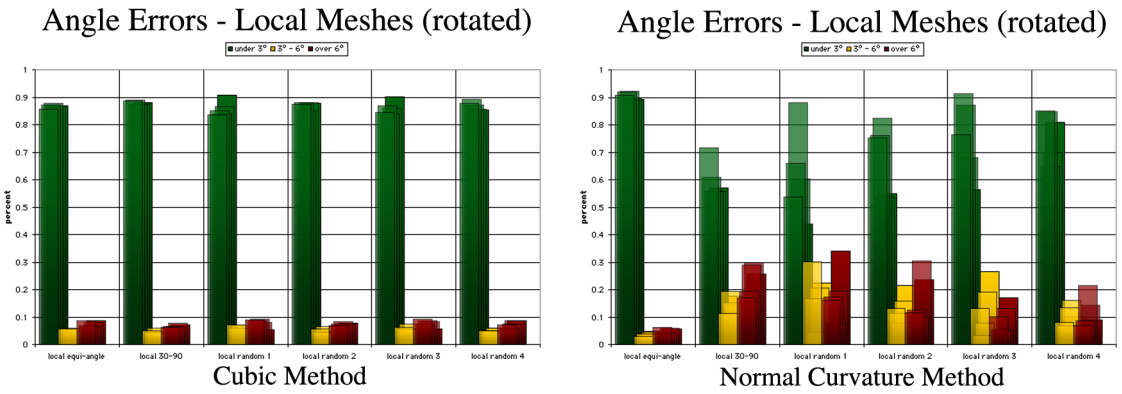

Figure 9. Distribution of errors in the approximated principal directions, found using five different rotations of the local meshes with each of the six local meshing schemes. Results are quantized into three ranges: less than 3° (green), between 3°-6° (yellow), and more than 6° (red). The graphs representing each of the different rotation conditions are layered in depth, with the zero rotation condition appearing at the rear.

Figure 10, also derived from Table 2, illustrates the effects of using higher order methods. The results are not qualitatively different from the cubic case. Overall, accuracy is good with these methods, at most points, but high errors remain at a significant fraction of the others.

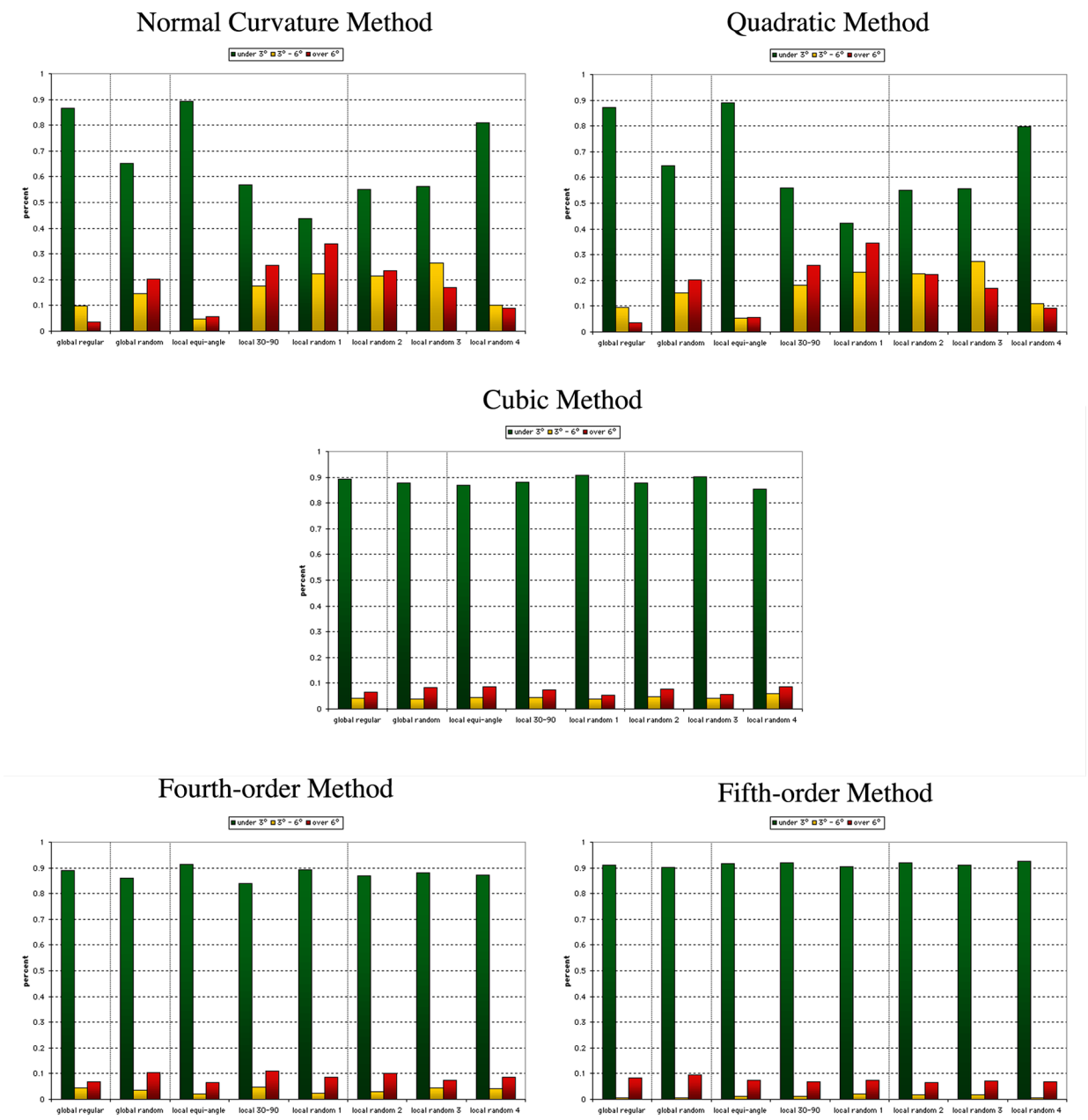

Figure 10. A comparison of the errors found in principal directions approximated using second through fifth order methods.

Figure 11, derived from Table 3 (also on the website), shows that the results are virtually identical when the constraint of equal edge length on the surface (as opposed to on the local parameterization in the tangent plane) is individually enforced for each local mesh. The local edge length was defined to be equal to the average of the lengths of the edges that had comprised the local mesh in the unequal edge length condition.

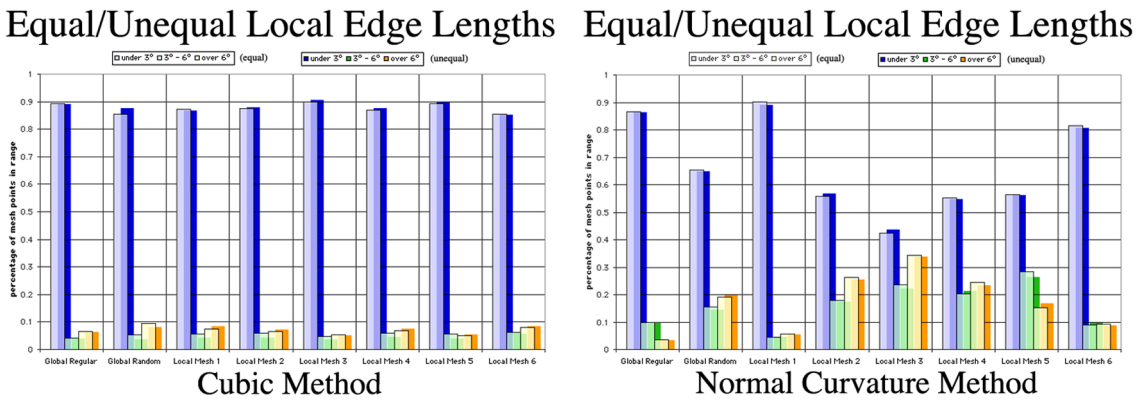

Figure 11. Distribution of errors in the approximated principal directions when edges in each local surface mesh are constrained to be of equal length in object space (foreground), as opposed to being equal length in parameter space (background).

Not shown here, but available in Table 4, are the results of further investigations in which we applied a *globally* equal edge length constraint, using two different edge lengths. As expected, we found that errors increased with increasing edge length. However the overall pattern of performance across the different meshes and methods remained qualitatively similar to what was found in the local equal (and unequal) edge length conditions.

IV.d The Effect of Errors in Surface Normal Approximations

Up to this point, we have been examining the performance of the various principal direction estimation algorithms using analytically-defined (error free) surface normals. In practice, of course, the true surface normal cannot be exactly determined, and it is useful to consider how errors in the surface normal computations can affect the accuracy of principal direction estimates achieved using each of the previously considered methods.

As several different surface normal approximation methods have been proposed, we decided to take a closer look at three:

1. The normalized average of the surrounding triangle unit normals.
2. The normalized average of the surrounding triangle unit normals, each weighted by the angle of the triangle at p.

3. The normalized average of the adjacent edges, each weighted by the sum of the cotangents of the angles opposite the edge in the two triangles sharing the edge [2]. This weighting only makes sense if each pair of triangles is acute, so another method must be used at vertices where a triangle is not acute.

We experimented with each of these schemes on our test surface, using the six local surface meshes described above, at the same 575 randomly chosen sample points. In all cases, the vertices that we used lay in the analytically-defined surface. Under these conditions, all of the normal approximation methods worked very well. The average error in the approximated normal over all mesh types was 0.90 degrees for the unweighted method, 0.97 degrees for the angle-weighted method, and 1.04 degrees for the edgelength-weighted method. The median error over all mesh types was 0.41 degrees for the unweighted method, 0.50 degrees for the angle-weighted method, and 0.54 degrees for the edgelength weighted method. Although the normal vector approximation errors in the tabulated data appeared to be superficially slightly smaller for the angle-weighted method in the cases of the global meshes and for the unweighted method in the cases of the local meshes, we do not believe that these differences are significant either practically or statistically.

Figure 12, derived from Table 5, summarizes the results of using approximated rather than exact surface normals in the principal direction estimation calculations for all three methods, pooling results from all five local mesh rotations. The approximated normals at each point were obtained by taking the normalized average of the surrounding triangle unit normals.

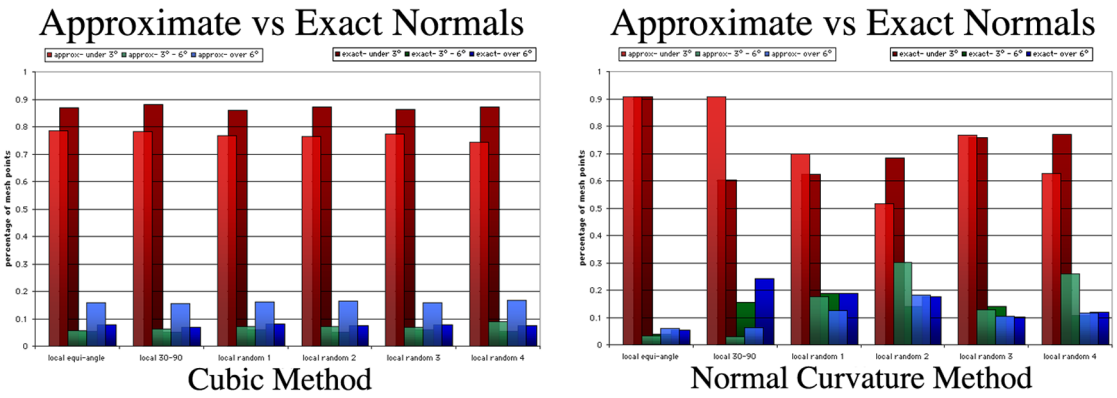

Figure 12. Distribution of errors in the approximated principal directions when computations are performed using approximated, as compared to exact, values for the surface normals. The results with the approximated normals are in the lighter colors, in the foreground, with the

results using exact normals given in a background layer for reference.

The main effect, which can be seen even more clearly in the tabulated data, seems to be an increase the amount of volatility in the accuracy of the estimates produced by the second order methods (for some meshes, in some rotations, resulting in greater errors, and for other meshes, or other rotations, resulting in fewer). Overall accuracy appears to be particularly good, for these methods, when the the approximated normals are used in combination with the two 'regular' local mesh schemes - the local equi-angle and the local 30-90, and not so good under the other mesh conditions. Consistent with our findings that the third and higher order methods appear to be less sensitive to variations in the local mesh characteristics, the impact on the accuracy of the estimates produced by the Cubic Method is more muted. There is a small but consistent downward trend in accuracy with the use of the approximated, rather than exact surface normal values, with the Cubic Method. However overall performance remains good, with over 75having errors of less than 3° and fewer than 20% having errors of greater than 6°.

In a last set of experiments, detailed in Table 6, we investigated the impact of reducing the valence at a vertex from 6 to 3. We computed this data for three different local meshes at 625 randomly selected points on a torus dataset. We also computed local meshes with valences 5 and 4 at the previously mentioned 575 selected points on the parametric surface dataset, and obtained similar results. We found, as might be expected, that the accuracy of the results produced by all three methods decreases as the valence drops, practically plummeting, for some meshes and methods, as the number of participating edges falls to 3. However the situation is rarely worse in the case of the cubic method than it is when either of the secod order methods is used.

Finally, we reexamined the our principal direction error findings considering all potentially 'near umbilic' sample points, defined as having with a principal curvature difference of less than 0.2, separately from the rest of the 575 points. In figure 13 we highlight the distribution of errors at these points in red, within the overall distribution. This explicitly reveals that large principal direction errors can occur **away from umbilical points**.

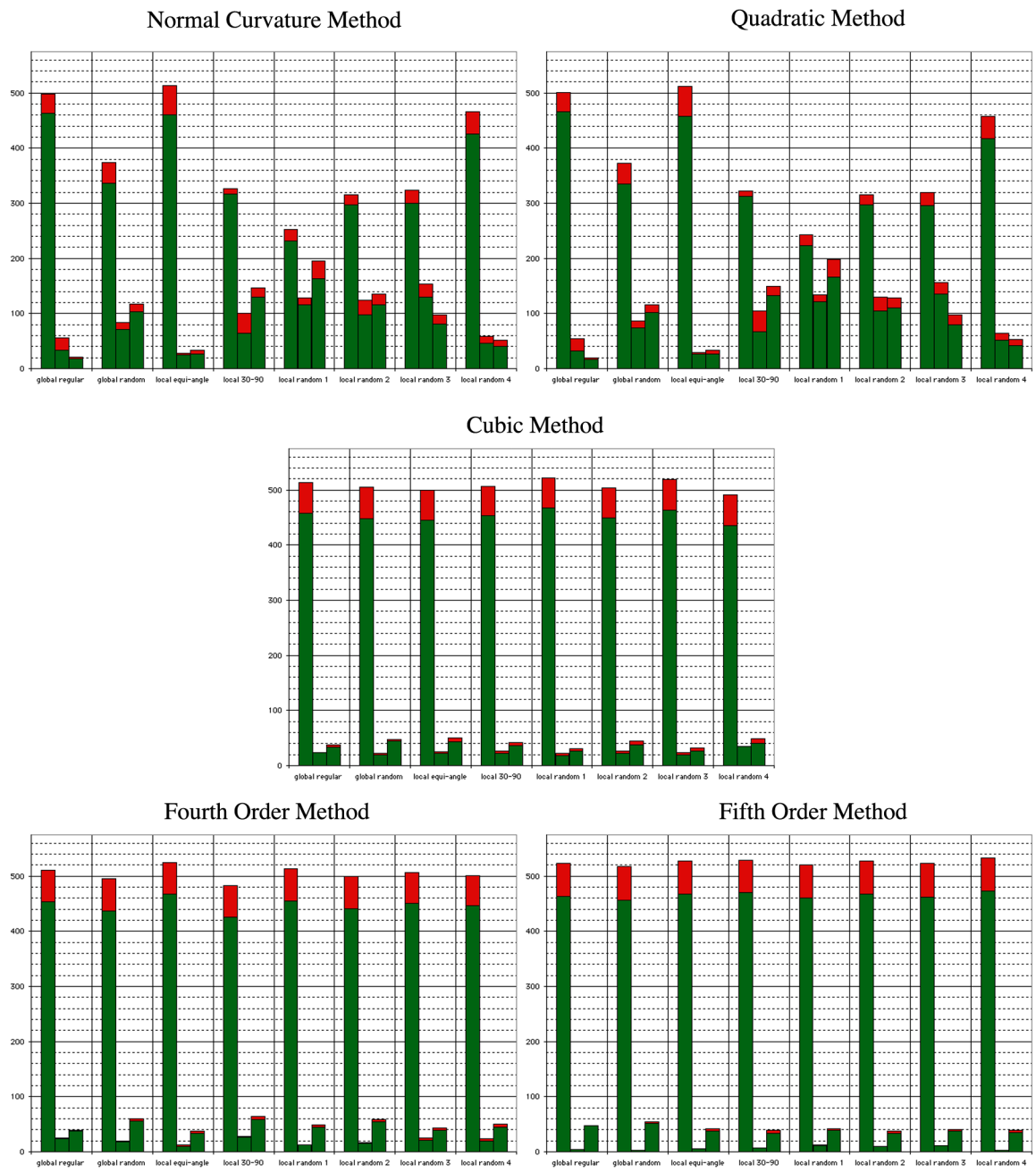

Figure 13. Histogram distribution of the angle error in the principal directions estimated using the five different methods and eight different mesh configurations discussed above, with the results due to 'near umbilic' points highlighted in red. As before, ranges for the bins are: less than 3°, between 3° and 6°, and greater than 6°.

In sum, our results suggest several conclusions:

1. The degree ≥ 3 methods nearly consistently outperform both of the degree 2 methods.
2. The regular meshes tend to produce better results than the random ones.
3. The equiangular mesh (Local Mesh 1) produces good results for all three methods.
4. Aligning edges with principal directions does not seem to matter.
5. Enforcing equality of adjacent edge length does not seem to matter, but the shorter all of the edges are the better.
6. Large errors can occur away from umbilical points.

After further mathematical investigation into potential sources of the errors encountered, we determined that in the case of the cubic method, the sizes of the errors in the estimates of the normal curvatures seems to be the factor most significantly influencing the size of the error in the principal direction estimates. In the case of the normal curvature method, the direction in which the errors were made seems, for some meshes, to be nearly as significant. In other words, the accuracy of the normal curvature method may be more dependent on the incidence of situations in which errors are able to cancel out.

V. Conclusions

We have shown that approximating principal directions on a surface mesh can be a tricky business. Although it is important to minimize the error in approximating various quantities such as normal curvature, an unfortunate pattern of small errors can produce large angle errors. Using a non-trivial test surface, we have shown that principal direction approximation methods are particularly prone to error when working with irregularly sampled data, and that these errors appear to be more severe for second order methods than for higher order methods that can take advantage of more of the known information about the surface.

There are many important directions for future work in this area. A major practical concern that we did not touch upon at all in our work is the problem of surface smoothing. In many cases, approximate meshes are obtained from a sampling process that is prone to a certain amount of intrinsic error. How might we best recover the mesh that describes the continuous smooth surface that is approximated by a collection

of noisy samples? If we have only a mesh that is highly irregular as well as noisy, how could we improve our chances of obtaining a smooth principal direction vector field and avoiding the introduction of large principal direction estimation errors? Would resampling to obtain a mesh in which the edge distribution is more equiangular help? Finally, it would be interesting to see how well the findings we have made hold up under further testing with a wider variety of mesh types and surface shapes.

REFERENCES

1. Xin Chen and Francis Schmitt, *Intrinsic Surface Properties from Surface Triangulation*, Proceedings, European Conference on Computer Vision, May 1992.
2. Mathieu Desbrun, Mark Meyer, Peter Schroder,and Alan H. Barr, *Discrete Differential Geometry Operators in nD*, preprint, July 22, 2000.
3. U. Diewald and M. Rumpf, *Visualization of Princial Curvature Directions by Anisotropic Diffusion*, 5th International Workshop on Visualization, Modeling and Visualization, 2000.
4. Bruce A. Duncan and Arthur J. Olsen, *Shape Analysis of Protein Surfaces*, Journal of Molecular Graphics, Vol. 10, No. 1, 1992.
5. Patrick J. Flynn and Anil K. Jahn, *On Reliable Curvature Estimation*, IEEE Computer Vision and Pattern Recognition, pp. 110-116, 1989.
6. Ahna Girshick, Victoria Interrante, Steven Haker and Todd Lemoine, *Line Direction Matters: an argument for the use of principal directions in line drawings*, First International Symposium on Non-Photorealistic Rendering and Animation, Annecy, 2000.
7. Gabriele Gorla, Victoria Interrante and Guillermo Sapiro, *Growing Fitted Textures on Surfaces*, IMA preprint, February 2001.
8. Stefanie Hahmann, *Visualization Techniques for Surface Analysis*, in Advanced Visualization Techniques, Chandrajit Bajaj, ed., John Wiley, 1999.
9. Bernd Hamann, *Curvature Approximation of 3D manifolds in 4D space*, Computer Aided Geometric Design, 11, pp. 621-632, 1994.
10. Aaron Hertzmann and Denis Zorin, *Illustrating Smooth Surfaces*, Proceedings of ACM SIGGRAPH 2000, pp. 517-526.
11. Victoria Interrante, *Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution*, Proceedings of ACM SIGGRAPH 97, pp. 109-116.
12. T. Maekawa, F.-E. Wolter and N. M. Patrikalakis, *Umbilics and Lines of Curvature for Shape Interrogation*, Computer-Aided Geometric Design, Vol. 13, No. 2, pp. 133-161, 1996.

13. 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, Vol. 17, No. 6, pp. 521-543, 2000.

14. Phillippe Samson and Jean-Laurent Mallet, *Curvature Analysis of Triangulated Surfaces in Structural Geology*, Mathematical Geology, Vol. 29, No. 3, pp. 391-412, 1997.

15. Pater T. Sander and Steven W. Zucker, *Inferring Surface Trace and Differential Structure from 3-D Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 6, pp. 833-854, September 1990.

16. Gabriel Taubin, *Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation*, Proc. 5th Intl Conf. on Computer Vision (ICCV'95), pp. 902-907, June 1995.