Journal for Geometry and Graphics
Volume 12 (2008), No. 1, 63–73.

Face-based Estimations of Curvatures on Triangle Meshes

Márta Szilvási-Nagy

Dept. of Geometry, Budapest University of Technology and Economics
H-1521 Budapest, Hungary
email: szilvasi@math.bme.hu

Abstract. Face-based curvature estimations on triangle meshes are presented in this paper. One method is based on the definition of osculating circles in normal planes, the other one directly on the definition of the curvature of curves. The regions of computation are disk neighborhoods with user specified radius constructed around each triangle of the mesh. The examples show reliable results in the estimation of the principal directions and of the selection of umbilical, flat, elliptic, hyperbolic and parabolic regions of the mesh also in such cases where vertex-based methods cannot be applied.

Keywords: triangle mesh, curvature, surface approximation
MSC 2007: 68U05, 68U07, 65020

1. Introduction

Our aim is to define new curvature values ordered to faces of a triangle mesh. By the help of circular disks introduced in [25] we give an estimation also for the Gaussian curvature, and with planar intersections a further estimation for normal curvatures. We extend and analyse the stability of our method given for estimating the principal curvatures and principal directions at an arbitrary point of the mesh also in such cases, where the known, frequently used methods do not work. Principal directions are important informations in several applications, e.g., for tool-path planning procedures in manufacturing.

Triangle meshes are frequently used approximations of free-form surfaces. They are represented by discrete data structures containing, e.g., the coordinates of three vertices and the coordinates of the face normal of each triangle. In geometric algorithms working on triangle meshes the description of topological informations are necessary. For this purpose different polyhedral data structures have been developed, which are vertex-oriented in most cases. Exceptions are few papers, where other representations are described as edge-oriented polyhedral data structure [23] or hierarchycal data structure developed for mesh simplification algorithms [12]. In vertex-based approaches a surface normal vector and curvature values are

ISSN 1433-8157/$ 2.50

defined from the data of the triangles in a well defined neighborhood of the actual vertex, and the results are proposed to be considered as estimated surface normal and surface curvatures, respectively at that point.

In a short survey of curvature estimation methods we mention the most frequently used ones and the crucial problems inspiring further investigations in the computation methods of surface normal vectors, normal, principal and Gaussian curvatures at a vertex point. All these values are influenced not only by the local shape of the mesh, but also by the chosen region, i.e., the set of triangles involved in the computation.

In curvature estimations there are basically two different approaches: discrete and analytical methods. Discrete methods work with discretized formulae of differential geometry developed for surface approximation and mesh smoothing ([5, 13, 15, 18, 19]). For example, the Gauss curvature at a vertex of the mesh is estimated with the help of the discretized Gauss-Bonnet theorem, the convergence of which has been investigated in [28]. Mean curvature is computed with the help of the discrete Laplace-Bertrami operator providing also methods for fairing irregular meshes. Analytical methods involve fitting a surface locally, then computing the curvatures of the fitted surface. Second order ([16, 20, 26, 27]) or third order [9] surfaces are used to approximate a part of the mesh. Five frequently used discretization schemes for approximating Gaussian curvature are compared in [7] with the result that the estimated curvatures are extremely sensitive to noise. The qualification of four other methods is similar in [20]. This explaines, why the huge number of papers on this field is still growing rapidly. A systematic survey of curvature estimation methods on triangular meshes is given in [8].

A basic question arises right at the beginning of the computation which region should be considered. A number of well known algorithms work on a one-ring neighborhood of a given vertex containing the triangles, one vertex of which coincides with the actual mesh vertex (Fig. 1).

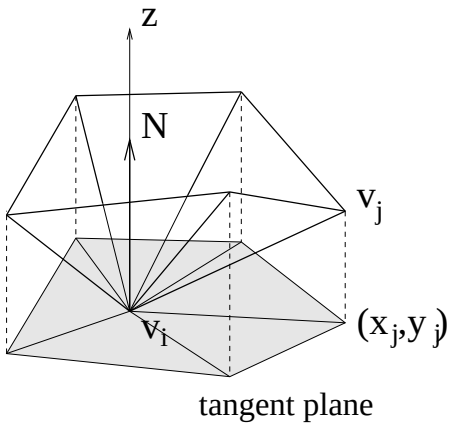

Figure 1: One-ring neighborhood of a vertex and the projection in the tangent plane

Of course, the results are depending on the shapes of the used triangles, and long, narrow triangles lead to huge fluctuations in the results of computations. The surface normal N at the point v_i is defined by a weighted average of the face normals in the one-ring neighborhood. The proposed weights are, e.g., areas or Voronoi surface areas or mixed surface areas of the triangles ([14, 16, 21]). Such an estimated normal vector determines the tangent plane and normal sections of the mesh. A widely used method for the computation of normal curvatures is to define the osculating circle by the surface normal and one edge emanating from the actual

vertex ([11, 15, 16, 26]). The curvature of this circle is (Fig. 2)

$$\kappa_n(v_i) \approx \frac{2 < N, (v_j - v_i) >}{|v_j - v_i|^2},$$

where $<,>$ denotes the dot product.

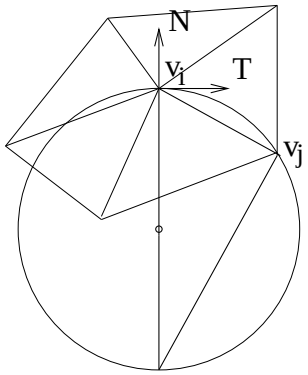

Figure 2: Osculating circle in one-ring neighborhood

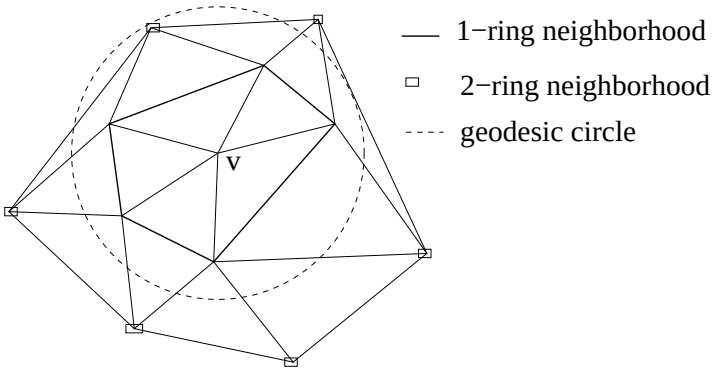

Figure 3: Two-ring and geodesic neighborhood of a vertex

Instead of one-ring neighborhoods, two-ring neighborhoods are used in [20] providing more informations about the surface approximated by the mesh. By least squares fitting of biquadratic Bézier patches better results are achieved also for irregular triangulations.

Geodesic circle neighborhood around a vertex is more independent from the shapes of the triangles (Fig. 3). In [17] geodesic neighborhoods are constructed, and normal curvatures are computed from the angle differences while moving the normal vector along a geodesic arc. Geodesic rings or their approximations are used also in [4, 6, 11, 22], then analytic methods are applied as a matrix method similar to TAUBIN's algorithm or computations using the Euler formula.

In the next vertex-based algorithm [21] first, a curvature value is defined for each face in an one-ring neighborhood of a vertex. The face-based curvatures are defined in terms of directional derivatives of the surface normals estimated at vertices, then a weighted average of face curvatures is computed at the vertex. A face-curvature is defined also in [2] from angles in the face and dihedral angles formed with the adjacent faces, then the maximum of those associated to triangles around a vertex is used for characterizing the mesh curvature at that point. Instead of one-ring neighborhood a spherical neighborhood is used in [1]. A weighted sum of dihedral angles within a sphere around the actual vertex normed by the corresponding mesh area defines the Gaussian curvature at this point.

The vertex-based algorithms work in vertex neighborhoods by collecting mesh informations around vertices. Obviously, they cannot be applied in such cases, where all the vertices are lying on boundaries or along feature edges (see Figs. 9 and 10). We propose a solution for this problem.

In this paper we define normal curvatures on each face of the triangle mesh in order to estimate principal directions and to characterize elliptic, umbilical, flat, parabolic and hyperbolic region (Section 2). In the examples (Section 3) we show the proposed method on "synthetic" and real triangulated surfaces.

2. Curvature values defined on a triangle face

Instead of computing surface properties at vertices in vertex neighborhoods we define curvature values associated to faces. The center of the defined region is the barycentric center of the given triangle. We intersect the mesh with normal planes passing through the face normal of the triangle, then we measure a given radius along the polygonal lines of intersection in both directions from the center point. In this way we get a number of curved diameters of the circular disk bent on the mesh around the face. We call this constructed disk geodesic circle (however, the normal sections are not geodesic curves) or "splat" after KOBBELT [13] (Fig. 4). Similar extended neighborhoods are defined around vertices by measuring a user specified radius along normal sections, and are called geodesic rings. The set of polygonal diameters of such a ring is called "spider" in [22].

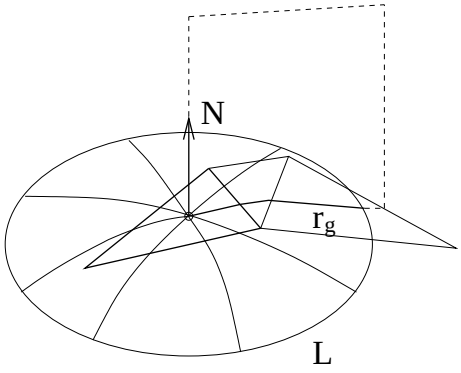

Figure 4: Geodesic disk around a face

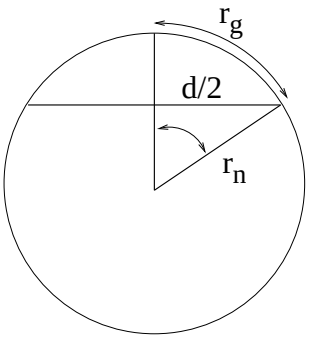

Figure 5: Osculating circle in a normal section

We define in each normal plane an osculating circle determined by the given radius and endpoints of the bent diameter and the face normal (Fig. 5) [25].

Denote r_g the given geodesic radius, d the chord length between the endpoints of the curved diameter, 2α the unknown central angle and r_n the required radius of the osculating circle. For the triangle and the arc belonging to the angle α

$$\sin\alpha = \frac{d}{2r_n} \quad \text{and} \quad r_g = r_n \cdot \alpha$$

hold. From these equations we get

$$\frac{1}{r_n} = \frac{2}{d}\sin\alpha \approx \frac{2}{d}\Big(\alpha - \frac{\alpha^3}{6}\Big) = \frac{2}{d}\Big(\frac{r_g}{r_n} - \frac{r_g^3}{6r_n^3}\Big), \quad 0 < \alpha << 1.$$

Consequently, the third order approximation of the normal curvature is

$$\kappa_n = \frac{1}{r_n} \approx \frac{1}{r_g}\sqrt{\Big(1 - \frac{d}{2r_g}\Big)\,6}\,. \tag{1}$$

Repeating this computation for a set of normal sections in the actual neighborhood of the given triangle, we obtain normal curvature values $\kappa_{n,i}$, $i = 1, \dots k$. If the mesh is a dense triangulation of a regular surface, then the normal planes belonging to the minimal and maximal normal curvatures are orthogonal to each other. They determine the principal directions. The plane of the shortest chord, where the computed curvature value is maximal,

determines the first principal direction T_1 and the first principal curvature κ_1 which is the normal curvature computed in this plane. This direction is fairly stable, even if we compute with smaller geodesic circles. The second principal direction T_2 is orthogonal to it. In the case of properly specified radius and nearly regular triangulation it is the direction belonging to the maximal chord length (Fig. 6). In the case of an umbilical or flat triangle (region) the minimal and maximal normal curvature values are equal (in the flat case zero), and each direction in the plane of the triangle is a principal direction.

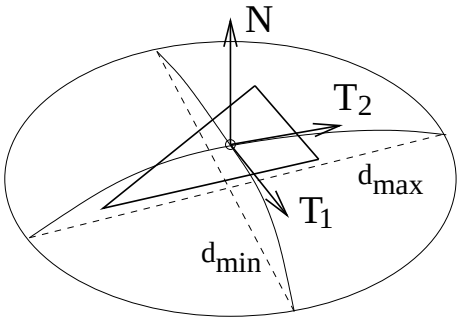

Figure 6: Principal directions

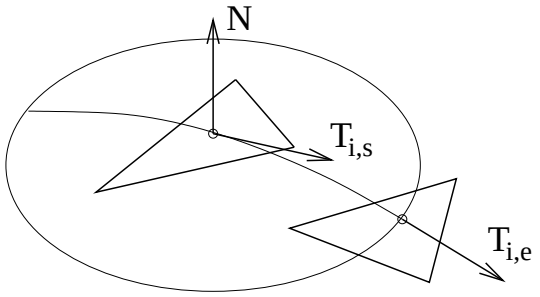

Figure 7: Tangent vectors of a normal section

We remark that the approximation of the normal curvature in formula (1) coincides with the curvature formula of Haantjes in [10]

$$\kappa_n^2 = 4! \lim_{l \to 0} \frac{l - d}{l^3},$$

where l is the arc length and d the chord length of the osculating circle, applied with the same geodesic radius $r_g = \frac{l}{2}$ and diameter d in the discrete form (i.e., without limit computation).

The other method for defining normal curvature in a given tangent direction is based directly on the curvature definition of space curves. We consider the directional vectors of the line of intersection in a normal plane with the mesh at the starting point and at the endpoint of this polygonal line. In Fig. 7 these vectors are denoted by T_{is} and T_{ie}, respectively. Instead of the limit

$$\kappa_{ni} = \lim_{\Delta s \to 0} \frac{\Delta \beta}{\Delta s},$$

where $\Delta\beta$ is the angle between the tangent vectors and Δs is the arc length between the two points, we define the normal curvature to the actual face in the direction of T_{is} by

$$\kappa_{ni} = \frac{\Delta \beta}{r_g}. \tag{2}$$

Here the radius r_g measures the length of the polygonal line between its starting and endpoint. This value is dependent on the radius, but if it is specified appropriately, the results correspond to those computed with the osculating circle.

We present a third curvature value, the Gaussian curvature ordered to a triangle face. The classical formula based on the Gauss-Bonnet theorem expresses the Gaussian curvature at a point v of an analytical surface using a geodesic circle in the following way [3]:

$$K(v) = \lim_{r_g \to 0} \frac{3}{\pi} \cdot \frac{2\pi r_g - L}{r_g^3}, \tag{3}$$

where r_g is the radius of the geodesic circle around the point v and L is the circumference (Fig. 4). The value of the Gaussian curvature is equal to the product of the two principal curvatures. $(K = \kappa_1 \cdot \kappa_2 = \max(\kappa_n) \cdot \min(\kappa_n))$

On the triangle mesh we compute the Gaussian face curvature not by the limit above, but by a circular disk around the barycentric center v of the triangle face. r_g is the length of each polygonal line in the normal sections (the length of the legs of the spider) and the circumference L is the length of the polygonal line consisting of the straight line segments between the endpoints of the adjacent polygonal diameters. In the examples we investigate the deviation of the product of the computed principal curvatures from this Gaussian curvature.

We have constructed an edge-oriented polyhedral data structure on the mesh which can differentiate inner, boundary, feature and silhouette edges, and is suitable for the computation of lines of intersection of a mesh with a plane.

3. Examples

In the examples we show so called synthetic and real meshes. A synthetic mesh is generated by a triangulation of the parameter domain of an analytical surface representation. Real meshes are generated from free-form surfaces triangulated by CAD systems (probably on the surface). In the analysis of the results we are focusing on the stability of the principal directions and on the classification of surface regions into flat, parabolic and elliptic types. The fluctuation in the results of our curvature estimations are comparable with the results of other algorithms in the literature.

The second and third columns in the Table 1 contain the proportional difference ($\max \kappa$ from (1)) between the maximal normal curvature and the value computed according to (1) and the Gaussian curvature K_1 computed from this normal curvature and the corresponding second principal curvature. The values in the next columns are the proportional errors of the maximal normal curvature ($\max \kappa$ from (2)) computed on the base of curvature definition by formula in (2) and the corresponding Gaussian curvature K_2. In the last column the Gaussian curvature K_3 is estimated by the discrete counterpart of the formula in (3). The value of r_g in the Table 1 is in fact a factor, which multiplies the "average size" of the actual triangle in order to get the radius of the geodesic disk. This factor is a more appropriate input value than the exact value of the radius due to different triangle sizes. The numerical representation of the mesh was restricted to 6 decimal digits. As the sign of the normal curvature depends on the orientation of the surface normal, we worked with its absolute value.

In Fig. 8 a synthetic mesh of a rotational cylinder with radius 10 (cyl1 in the Table 1) is shown. The radius factor of the disk is 3. According to our investigations, the best results have been achieved with radius factors between 2.5 and 3.5. The actual triangle (as each point on the cylinder) is of parabolic type. The longer straight line segment is showing into the first principal direction belonging to the maximal normal curvature. Because the mesh is regular, all these computed curvatures are equal for each triangle.

In Fig. 9 a coarse synthetic mesh of the same cylinder is shown (cyl2 in the Table 1). Of course, the radius factor must be less then 1. Despite of the bad shape of the triangles, the computed results, especially the curvatures computed by the osculating circle and the Gaussian curvatures are very good. This kind of triangulation cannot be handled by vertex-based computations.

In Fig. 10 a real mesh of a cylinder is shown with a geodesic disk on the upper face (cyl3 in the Table 1). The triangles on this face are of flat type, and each geodesic disk lying inside

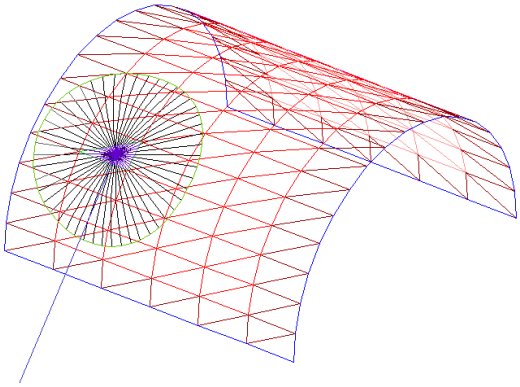

Figure 8: Geodesic disk on a cylinder (cyl1)

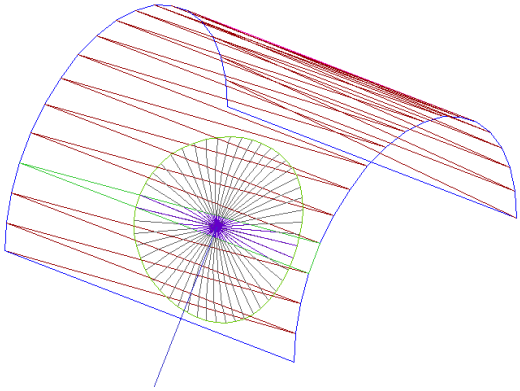

Figure 9: The same cylinder (cyl2)

of the upper face and all the three computation methods have given very accurate results. Such triangulations, from boundary to boundary, occur frequently in flat regions, especially in STL files. The mesh has two four-sided holes on the side, the boundary edges of which are registered in the data structure [24].

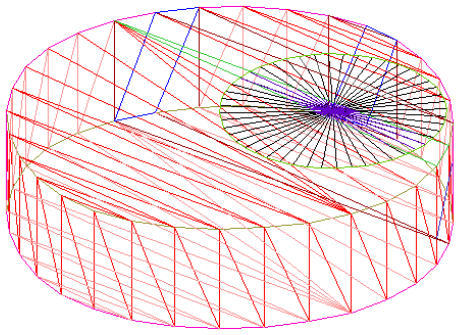

Figure 10: A real mesh of a cylinder (cyl3)

In Fig. 11 a part of a torus is shown (tor1 in the Table 1) and a geodesic disk on it. The maximal normal curvature equals to the reciprocal of the radius of the meridian circle at each point of the torus. The investigation of the variation of the Gaussian curvature K_3 computed on the base of the Gauss-Bonnet theorem concluded that larger geodesic disks led to better results with this formula on convex (elliptic) surfaces. The computed values are constant along parallel circles according to the shape of the torus.

In Fig. 12 a coarser, irregular mesh of the same torus is shown. Here bigger disks are necessary for the same accuracy in the results than in the denser triangulation. (The regular mesh is not shown, the computed values are in the row tor2 in the Table 1.) Fig. 12 shows also the influence of the triangulation. The parameter values are disturbed randomly by 3%. The results show the same relative error in the computations with the formulae (1) and (3) as in the regular case. Computing with the tangent vectors by the formula (2) is less accurate in this case. The results for irregular meshes are shown in the Table 1 in the rows tor3 in the dense and tor4 in the coarse triangulation, respectively.

Noise added to the coordinate values (i.e., the mesh points are not l ying on the torus) lead to bigger errors (3% random change in the coordinates course 16-29% error).

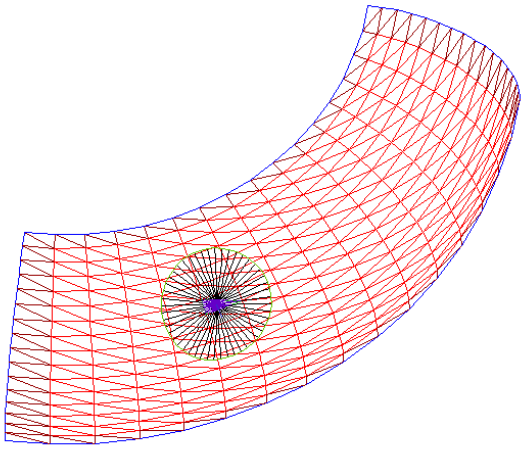

Figure 11: Regular mesh of a torus (tor1)

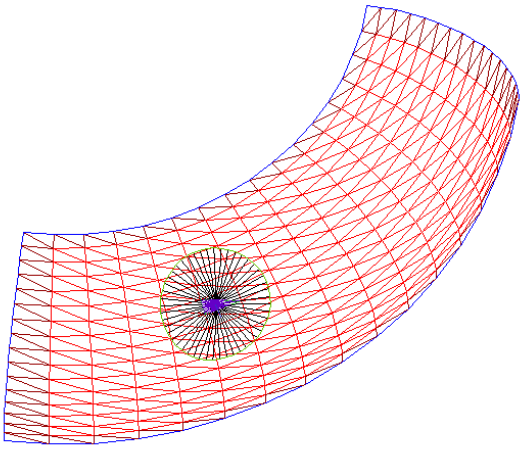

Figure 12: Irregular coarse mesh of the same torus (tor4)

Table 1: Data of examples

model, r_g	$\max\kappa$ from (1)	K_1	$\max\kappa$ from(2)	K_2	K_3 from (3)
cyl1, 3	1.4%	0.0%	1.4%	0.0%	0.3%
cyl2, 0.7	0.4%	0.0%	1.1%	0.0%	0.2%
cyl3, 0.5	0.0%	0.0%	0.0%	0.0%	0.0%
tor1, 3	0.5%	1.5%	1.2%	7.0%	1.2%
tor2, 4.5	0.9%	2.0%	0.4%	2.0%	1.2%
tor3, 4.0	0.5%	1.5%	1.2%	8.0%	1.2%
tor4, 4.6	0.9%	2.7%	0.4%	4.3%	1.9%

In the construction of a geodesic disk its center point has been set into the barycentric center of the actual triangle. The next example shows that the maximal normal curvature and the corresponding principal direction are very stable independently from the exact position of the center point in the actual triangle. This is demonstrated by stepping in the principal direction by the given radius on the mesh (Fig. 13). The path of this movement is a meridian circle of the torus, however, some center points are close to a triangle edge. The bordering polygonal line of the disks are computed and drawn with 24 diameters. The position of the center point in the actual triangle is shown in an enlarged part of the path (Fig. 14). The straight line segments are pointing into the principal directions.

We emphasize that in cases shown in Figs. 9 and 10 vertex-based methods cannot be applied. Our method applying circular disks is less sensitive on the irregular triangulations than vertex-based computations, and especially the principal direction belonging to the maximal normal curvature is very stable.

4. Conclusion

We have introduced curvature values ordered to faces in triangle meshes by laying a flexible circular disk with user-specified radius onto each face of the mesh. From the chords of such a

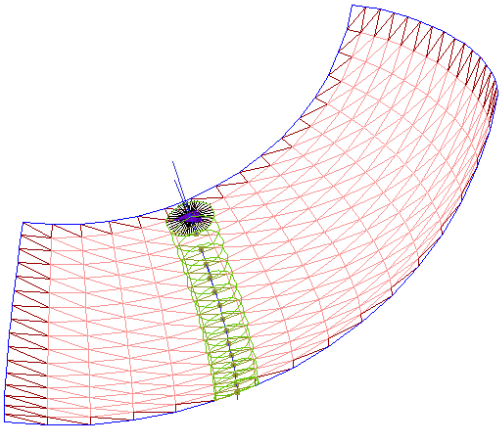

Figure 13: Moving in the computed principal directions

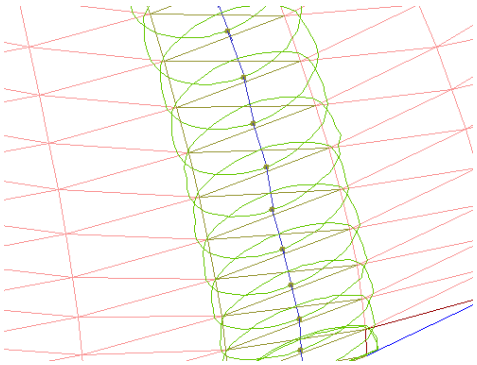

Figure 14: The center points of the disks are not in the barycenters of the faces

bent disk and on the base of classical curvature definitions we have defined normal, principal and Gaussian curvatures. Novel in the presented algorithms is the characterization of the mesh by curvature values and principal directions associated to mesh triangles. Our method has the advantages compared to former estimations that it does not use estimated normal vectors, and the defined osculating circle approximates that of the underlying surface in third order. Moreover, our method can be used in regions with long, narrow triangles and in regions without inner mesh vertices where vertex-based methods don't work. The examples have shown that the obtained principal curvature values and the corresponding principal directions are quite reliable, when the radius of the disk achieves an optimal size. Our method provides a good classification of elliptic, parabolic, flat and hyperbolic regions of the mesh.

We have implemented the presented algorithms in Java on PC.

Aknowledgement

This work was supported by a joint project between the TU Berlin and the BUTE and by the Hungarian National Foundation OTKA No. T047276.

References

[1] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, M. Desbrun: *Anisotropic polygonal remeshing*. ACM Trans. Graph. SIGGRAPH'2003 Conf. Proc. **22(3)**, 485–493 (2003).

[2] E. Béchet, J.-C. Cuillire, F. Trochu,: *Generation of a finite element MESH from stereolithography (STL) files*. Computer-Aided Design **34**, 1–17 (2002).

[3] M.P. Do Carmo: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[4] X. Chen, F. Schmitt,: *Intrinsic surface properties from surface triangulation*. In: Proceedings of the European Conference on Computer Vision 1992, pp. 739–743.

[5] M. Desbrun, M. Meyer, P. Schröder, A.H. Barr: *Implicit fairing of irregular meshes using diffusion and curvature flow*. In: Computer Graphics Proceedings, SIGGRAPH'99 pp. 317–324.

[6] S.S. Dong, G.Z. Wang: *Curvatures estimation on triangular mesh*. Journal of Zhejiang University SCIENCE **6A**/Suppl. I., 128–136 (2006).

[7] P.J. Flynn, A.K. Jain: *On reliable curvature estimation*. In: IEEE Conf. Comput. Vision Patt. Recogn. 1989, pp. 110–116.

[8] T.D. Gatzke, C.M. Grimm: *Estimating curvature on triangular meshes*. International Journal of Shape Modeling, **12**/1, 1–28 (2006).

[9] J. Goldfeather, V. Interrante: *A novel cubic-order algorithm for approximating principal direction vectors*. ACM Transactions on Graphics **23**/1, 45–63 (2004).

[10] J. Haantjes: *Distance Geometry. Curvature in abstract metric spaces*. Proc. Akademie van Wetenschappen, Amsterdam **50**, 496–508 (1947).

[11] E. Hameiri, I. Shimshoni: *Estmating the principal curvatures and the Darboux frame from real 3D range data*. Computers & Graphics **28**, 801–814 (2004).

[12] H. Hoppe: *Progressive meshes*. Computer Graphics Proceedings, Annual Conference Series 1996, pp. 99–107 .

[13] L. Kobbelt, M. Botsch: *A survey of point-based techniques in computer graphics*. Computers & Graphics **28**, 801–814 (2004).

[14] N. Max: *Weights for computing vertex normals from facet normals*. Journal of Graphics Tools **4**, 1–6 (2000).

[15] M. Meyer, M. Desbrun, P. Schröder, A.H. Barr: *Discrete differential-geometry operators for triangulated 2-manifolds*. In: H.C. Hege, K. Polthier (eds.:) *Visualization and Mathematics III*. Springer Verlag, 2003, pp. 35–57.

[16] H.P. Moreton, C.H. Séquin: *Functional optimization for fair surface design*. In: SIGGRAPH'92 Conference Proceedings, pp. 167–176.

[17] D.L. Page, Y. Sun, A.F. Koshan, J. Paik, M.A. Abidi: *Normal vector voting: Crease detection and curvature estimation on large, noisy meshes*. Graphical models **64**, 199–229 (2002).

[18] U. Pinkall, K. Polthier: *Computing discrete minimal surfaces and their conjugates*. Experimental Math. **2**, 15–36 (1993).

[19] K. Polthier, M. Schmies: *Straightest geodesics on polyhedral surfaces*. In K. Polthier, H.C. Hege (eds.): *Mathematical visualization*, Springer Verlag, 1998, pp. 391–409.

[20] A. Razdan, M.S. Bae: *Curvature estimation scheme for triangle meshes using biquadratic Bézier patches*. Computer-Aided Design **37**, 1481–1491 (2005).

[21] S. Rusinkiewicz: *Estimating curvatures and their derivatives on triangle meshes*. Proc. 3DPVT'04, pp. 486–493.

[22] P. Simari, K. Singh, H. Pedersen: *Spider: A robust curvature estimator for noisy, irregular meshes*. Technical report CSRG-531, Dynamic Graphics Project, Dept. of Comp. Sci., Univ. Toronto, 2005.

[23] M. Szilvási-Nagy, Gy. Mátyási: *Analysis of STL files*. Mathematical and Computer Modelling **38**, 945–960 (2003).

[24] M. Szilvási-Nagy: *Removing errors from triangle meshes by slicing.* Third Hungarian Conference on Computer Graphics and Geometry, Nov. 17–18, 2005, Budapest, pp. 125–127.

[25] M. Szilvási-Nagy: *About curvatures on triangle meshes.* KoG **10**, 13–18 (2006).

[26] G. Taubin: *Estimating the tensor of curvature of a surface from a polyhedral approximation.* In: ICCV'95 Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, Washington DC, 902, 1995.

[27] P.H. Todd, R.J.Y. McLeod: *Numerical estimation of the curvature of surfaces.* Computer-Aided Design **18**, 33–37 (1986).

[28] G. Xu: 2006. *Convergence analysis of a discretization scheme for Gaussian curvature over triangular surfaces.* Comput.-Aided Geom. Design **23**, 193–207 (2006).

Received January 15, 2008; final form May 5, 2008