Original scientific paper
Accepted 30. 11. 2010.

MÁRTA SZILVÁSI-NAGY*

Surface Patches Constructed from Curvature Data

Surface Patches Constructed from Curvature Data

ABSTRACT

In this paper a technique for the construction of smooth surface patches fitted on triangle meshes is presented. Such a surface patch may replace a well defined region of the mesh, and can be used e.g. in re-triangulation, mesh-simplification and rendering.

The input data are estimated curvature values and principal directions computed from a circular neighborhood of a specified triangle face of the mesh.

Key words: triangle mesh, principal curvatures, local surface approximation

MSC 2010: 65D17, 68U07, 65D18, 68U05

Dijelovi plohe konstruirani iz podataka o zakrivljenosti

SAŽETAK

U ovom članku se prikazuje metoda za konstrukciju glatkih dijelova plohe koji odgovaraju mreži trokuta. Takav dio plohe može zamijeniti dobro definirano područje mreže, i može se upotrijebiti npr. u retriangulaciji, simplifikaciji mreže i renderiranju.

Ulazni podaci su procijenjene vrijednosti zakrivljenosti i glavni smjerovi izračunati iz kružne okoline određenog trokuta mreže.

Ključne riječi: mreža trokuta, glavne zakrivljenosti, lokalna aproksimacija plohe

1 Introduction

Triangle meshes are the most common surface representation in many computer graphics applications. In order to reduce a model's size, cut the storage space and decrease the time required to display the model, simplification algorithms are applied to the mesh. A simplification algorithm in [5] is based on iterative edge contraction analyzed on a tessellation of twice-differentiable surfaces. For local approximation ellipsoids at the mesh vertices are computed that characterize the local shape of the surface. These quadrics are elongated in directions of low curvature and thin in directions of higher curvature. Edge contractions are determined by minimizing the defined quadric metric which result in appropriately stretched triangles in the simplified mesh.

Also in point-based surface models some approximating geometric primitives are used for a compact representation. Piecewise constant surfaces in the form of circular discs, so-called splats have been proposed for rendering purposes in [8]. Splat radii depend on local curvature properties. From estimated normals at data points a smoothly varying normal field is generated for each patch, which is necessary for producing photo-realistic results in ray-tracing methods. For optimal local approximation to the underlying surface elliptical splats have been used in [7]. The two axes are aligned to the principal curvature directions and the radii are inversely proportional to the minimum and maximum curvatures. From sampled points of a surface analytic patches are constructed in a polynomial form in [6]. First, the principal curvatures and the Darboux frame are estimated using sampled points along three curves passing through the point of interest. The coefficients of the polynomial function describing the required patch are computed by a constrained surface fitting method using the total Gaussian curvature. A different type of surface elements, so-called surfel objects are used in [10]. Surfels are point primitives with attributes as surface normal, position, orientation and texture. They allow the creation of 3D graphic models with complex shapes.

Several methods have been developed for getting curvature information from discrete surface representations. A survey of results in discrete differential geometry and flexible tools to approximate important geometric attributes, including normal vectors and curvatures on arbitrary triangle meshes, also applications such as mesh smoothing

* Supported by a joint project between the TU Berlin and the BUTE.

are given in [9]. The quadratic paraboloid is the typical analytic regular surface used for local approximation of a mesh, which is usually computed by least squares method ([3]). The principal curvatures of the underlying surface are estimated by the principal curvatures of such a paraboloid. In [2] the quadric is extended by linear terms, and is fitted iteratively by correcting the surface normal, which leads to a new quadric in each step. The curvature estimation method in [11] uses biquadratic Bézier patches as a local fitting technique. One advantage in using parametric form of the locally approximating surface is the ability to add smoothing constraints when dealing with noisy data. A surface-based method is applied to point-based surfaces in [19]. The moving least-squares method is used to compute analytic surfaces locally fitting a point cloud, which provides direct curvature evaluation but also feature recognition and rendering applications.

Instead of fitting a smooth surface to a local set of points a different approach estimates the curvatures directly from the discrete data of the triangulation. A classical paper of Taubin ([17]) shows a normal-curvature based method. First, the normal curvatures are estimated in the direction of each edge leaving the mesh vertex of interest, then the second fundamental tensor is computed. The principal curvatures are determined as eigenvalues of this tensor defined at each vertex. The algorithm in [1] computes the normal curvatures by fitting circles at each vertex in three (or more) tangent directions using two neighboring points and applying the Meusnier theorem. Then the principal curvatures are computed on the base of Euler formula. A modification of this algorithm is shown in [4] which is adjusted to deal with real discrete noisy range data. A finite-differences approach for estimating curvatures on irregular triangle meshes is presented and used for computing derivatives of curvature and higher-order surface differentials in [12].

The objective of this work is to view a surface represented by a triangle mesh as a collection of local patches. In our algorithm a circular or elliptical surface patch is constructed around a specific triangle face of the mesh, and is represented by a trigonometric vector function. The input data are estimated curvature values and principal directions computed from a neighborhood of the given triangle in the following way ([14], [15]).

First a circular neighborhood is constructed around a specified triangle (Fig. 1). This bent disc laid onto the mesh is determined by a given arc length as "radius" r_g (see Fig. 2) which is measured in the normal sections of the mesh along the polygonal section lines. From the chord length of a "diameter", denoted by d, the radius r_n of the osculating circle in the actual normal plane is computed by a third order approximation method. By repeating this computation in sufficient many normal planes, curvature values can be ordered to the actual triangle face. The maximal and minimal diameters determine the principal directions T_1, T_2 and the curvature radii (Fig. 3) denoted by ρ_1 and ρ_2 in Fig. 4.

This construction and curvature estimation work also on triangle meshes, where the usual vertex-based methods cannot be applied. Such a mesh is shown in Fig. 7.

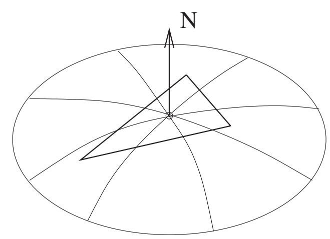


Figure 1: *Circular neighborhood of a face*

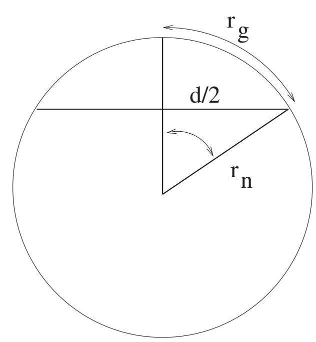


Figure 2: *The osculating circle in a normal plane*

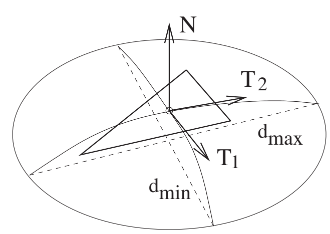


Figure 3: *Principal directions*

2 The approximating surface patch

Having the principal directions T_1 and T_2 at the barycenter P of the actual triangle face, a local basis is defined by $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\} = \{T_1, T_2, N\}$ (Fig. 3). In the normal plane containing the first principal direction the osculating circle has the radius ρ_1, the central angle of the given arc length is $\alpha = r_g/\rho_1$, $\rho_1 \neq 0$. In the second normal plane determined by the second principal direction, orthogonal to the first one, the curvature radius is ρ_2 and the central angle of the given arc length is $\beta = r_g/\rho_2$, $\rho_2 \neq 0$ (Fig. 4). An arbitrary point of the circles is Q_1 and Q_2, respectively. The position vector of these points, i.e. the vector equations of these circle arcs in the local coordinate systems are

$$\vec{PQ_1} = \rho_1 \sin\vartheta \mathbf{e}_1 + (-\rho_1 + \rho_1 \cos\vartheta)\mathbf{e}_3,\ 0 \leq \vartheta \leq \alpha$$

for a negative principal curvature $\kappa_1 = -1/\rho_1$ and

$$\vec{PQ_2} = \rho_2 \sin(\vartheta\frac{\beta}{\alpha})\mathbf{e}_2 + (\rho_2 - \rho_2 \cos(\vartheta\frac{\beta}{\alpha})\mathbf{e}_3,\ 0 \leq \vartheta \leq \alpha,$$

when the principal curvature $\kappa_2 = 1/\rho_2$ is positive. In Fig. 4 $\vartheta^* = \vartheta(\beta/\alpha)$.

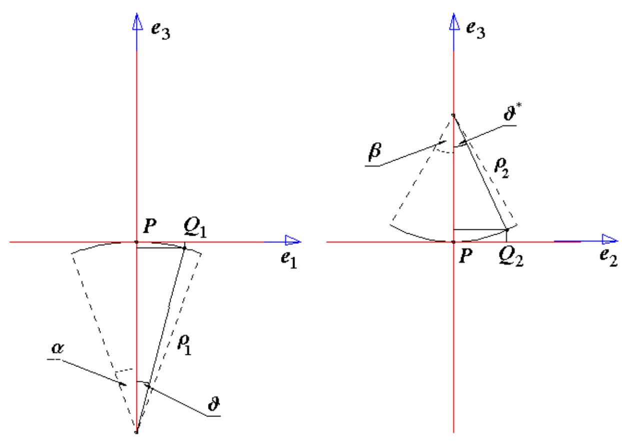


Figure 4: *Arcs of osculating circles in the principal normal sections of given length $2r_g$*

These circle arcs will determine the approximating surface patch around the point P. If they are of the same arc length, a circular patch will be generated.

In the construction of an elliptical patch we assume that $\rho_1 < \rho_2$, i.e. the neighborhood of the point P is more flat in the second principal direction. Then we enlarge the arc length of the osculating circle, in this way the patch to be generated will be stretched in this direction. If $\kappa_2 \to 0$, e.g. for a cylinder, then $\rho_2 \to \infty$ and $\beta \to \pi/2$. Consequently, the corresponding arc is a straight line segment of length $\to \infty$. This arc length should be limited in the construction. In our examples this limit has been set to $2.5 \cdot r_g$, and $\beta = min\{\alpha, 2.5 \cdot (r_g/\rho_2)\}$, $\rho_2 \neq 0$, while for circular patches $\beta = r_g/\rho_2$.

The vector equation of the surface patch is generated from the two circle arcs in the normal planes through the principal directions in such a way that the third coordinate of the arbitrary point of the patch is a convex combination of those of the points Q_1 and Q_2, while the normal plane is rotating around the normal vector with the angle φ.

$$\mathbf{q}(\vartheta,\varphi) = \mathbf{p} + \rho_1 \sin\vartheta \cos\varphi \mathbf{e}_1 + \rho_2 \sin(\vartheta\tfrac{\beta}{\alpha}) \sin\varphi \mathbf{e}_2 + \\ \left((-\rho_1 + \rho_1\cos\vartheta)\cos^2\varphi + \ (\rho_2 - \rho_2\cos(\vartheta\tfrac{\beta}{\alpha})\sin^2\varphi\right)\mathbf{e}_3, \\ 0 \leq \vartheta \leq \alpha,\ 0 \leq \varphi \leq 2\pi.$$

In this equation the neighborhood of the point P is assumed to be of hyperbolic type. In the case of an elliptic neighborhood the third coordinates of the arc points Q_1 and Q_2 have to be computed by the same formula. In Fig. 5 the boundary curve of the generated surface patch is shown in a hyperbolic neighborhood of the point P.

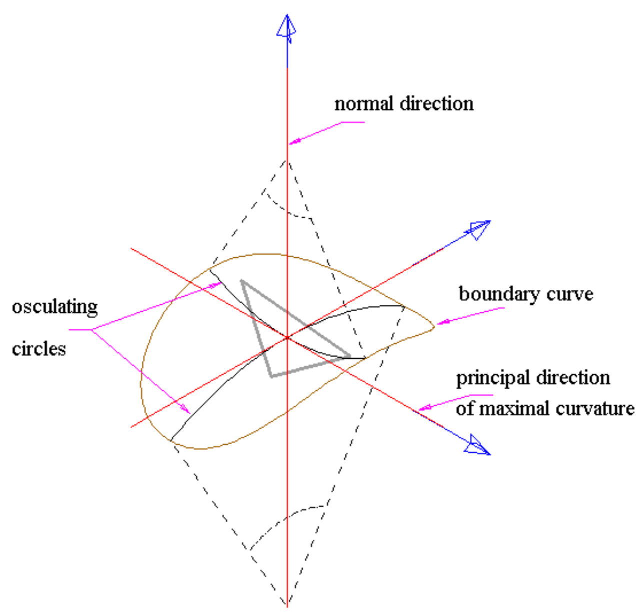


Figure 5: *The boundary curve of the surface patch*

If the neighborhood of the point P is of elliptic type, the constructed patch is situated tangential on the concave side of the mesh. Assuming that the mesh vertices are lying on the underlying surface approximated by the mesh, the position of this "inscribed" surface patch will be corrected by translating it in the normal direction. The measurement of this translation is computed as the mean value of the distances of the three vertex point of the base triangle to the generated patch.

3 Examples and analysis of the patch construction

In the examples circular and elliptical patches on "synthetic" and "real" meshed surfaces have been constructed. Synthetic meshes are generated by computing the mesh points from the vector equation of the surface on the corresponding triangulated parameter domain. Real meshes are chosen from an available collection offered for test purposes.

Computations in a normal plane require an appropriate polyhedral data structure adapted to the triangle mesh. Modeling systems usually store tesselated surfaces in STL format, which is a set of triangle faces, each described by three vertex points and a normal vector perpendicular to the face. In the implementation of our algorithms the STL data have been scanned for searching identical edges, hereby adjacent faces, and a polyhedral data structure based on a doubly linked edge system has been generated [13]. The polygonal line of intersections of the normal planes and the mesh, the estimation of normal curvatures and principal directions have been computed by the help of this data structure implemented in Java programming language. Then the mesh and the results of these computations have been transferred to the symbolical algebraic program package Mathematica [18]. The analytical description of the constructed patches, the error estimation and the figures have been made by Mathematica. The distance between a vertex point and the surface patch (Fig. 6) has been computed also by Mathematica.

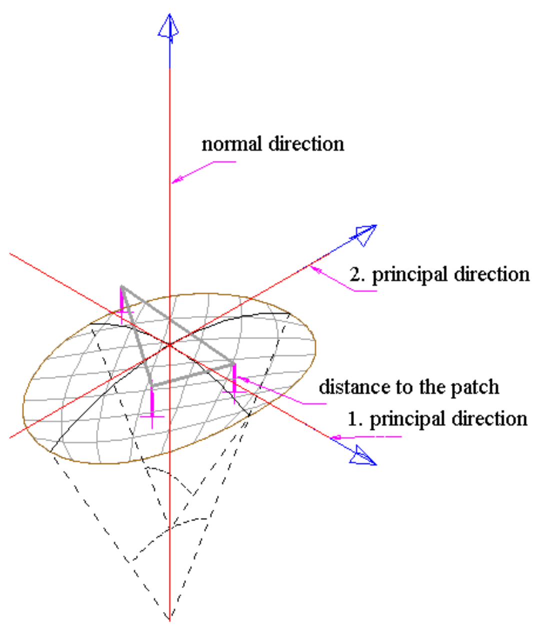
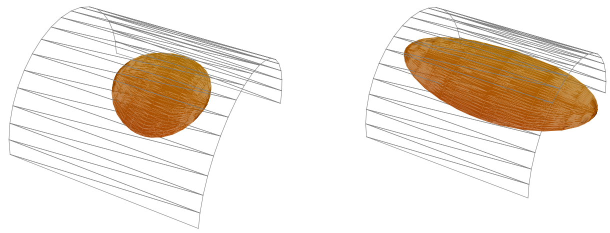


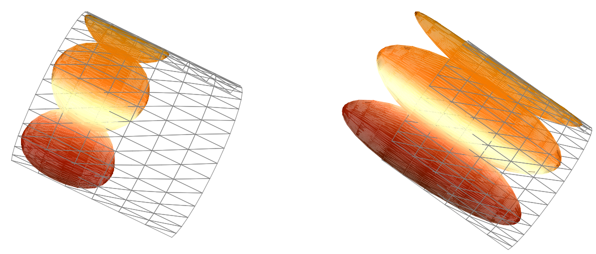
Figure 6: *Translation of the surface patch*

In Fig. 7 a synthetic mesh of a half cylinder is shown. In this triangulation all the mesh points are lying on the boundary circles, and no vertices are in the region of the construction. In such a case the vertex based algorithms do not work, but our face based curvature estimation resulted in accurate principal curvatures and principal directions ([14], [15]). The circular and elliptical patches are computed from these data. Series of the constructed patches may replace specified parts of the mesh (see Fig. 8).



a b

Figure 7: *Circular and elliptical patch on a cylindrical meshed surface*



a b

Figure 8: *Series of patches on a dense mesh of the cylinder.*

The constructed patch approximates the mesh in a neighborhood of its center point. The approximation error is depending on the size of the constructed patch. Working on an analytic surface given by a vector equation, the input data of the patch construction, namely the principal directions and curvatures at the actual surface point are computed from the surface equation [16]. On triangle meshes these data are estimated values. In both cases the osculating circles in the principal directions approximate the corresponding normal sections of the meshed surface with an error. This error can be easily calculated at the end points of the circle arc. In this way the arc length r_g determining the size of the generated patch can be given using this error value as prescribed tolerance. The question is, how large is the difference between the patch boundary and the mesh in other normal sections. This difference has been computed by the distance of a testpoint on the boundary curve of the patch to the mesh. As the triangle faces lying in the actual region of the construction are registered in the polyhedral data structure, the nearest face to the testpoint is selected for computing the error (Fig. 9).

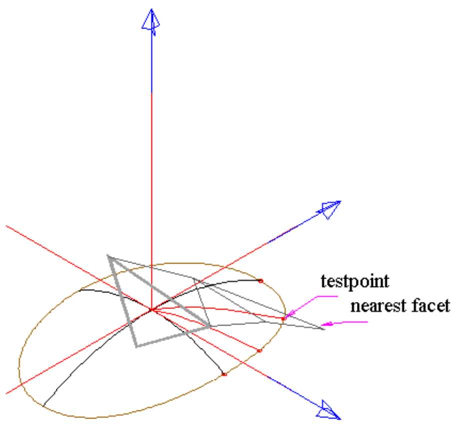
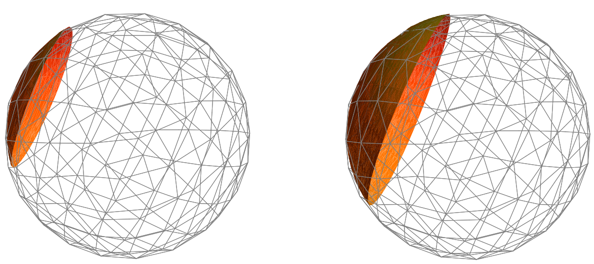


Figure 9: *Error estimation*

On a real mesh of the sphere (Fig. 10) four testpoints have been chosen on a quarter of the patch boundary, i.e. in the interval $\varphi \in [0, \pi/2]$. The relative errors to the given arc length of the osculating circle are for the smaller patch in turn 0.18%, 0.06%, 0.04%, 0.65% and for the larger patch accordingly 0.03%, 0.8%, 1.7%, 2.5%. For the smaller patch the order of these errors roughly equals to the approximation error of the input data. The error is growing fast by increasing the size of the patch. In our implementation an "optimal siz" of the patches has been chosen interactively.

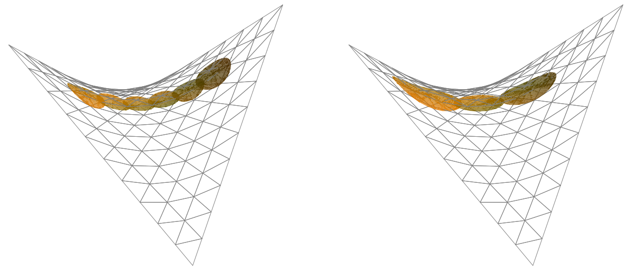


a b

Figure 10: *Circular patches on a "real" mesh of a sphere*

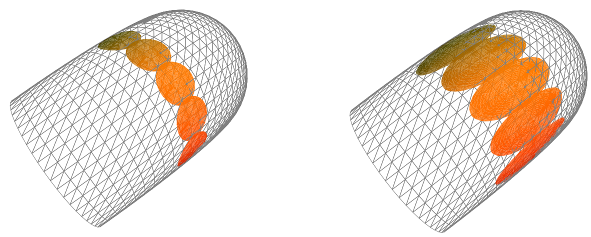
The measurements of the covered regions by circular and elliptical patches have been investigated on synthetic meshes of a saddle surface (Fig. 11) and a torus (Fig. 12).

In the case of "real objects" the size of approximating patches may vary strongly. A large elliptical patch is shown in Fig. 13.



a b

Figure 11: *Circular and elliptical patches on a saddle surface*



a b

Figure 12: *Circular and elliptical patches on a torus*

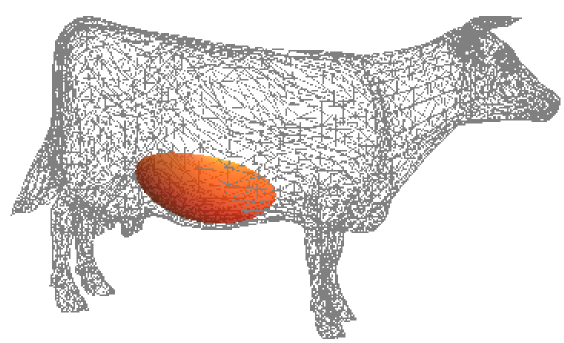


Figure 13: *Elliptical patch on the mesh of the cow*

4 Conclusions

A construction method for generating circular and elliptical surface patches has been presented for local approximation of triangle meshes. This is a novel, direct way for building a vector function of such analytical surfaces from estimated curvature data. The numerical analysis has been made on synthetic meshes of cylindrical and toroid surfaces. The implementation of the algorithm has been developed in Java programming language and by the help of the symbolical algebraic program package Mathematica.

References

[1] X. Chen, F. Schmitt, Intrinsic surface properties from surface triangulation. In *Proc. European Conf. Comp. Vision* 1992, 739–743.

[2] R.V. Garimella, B.K. Swartz, Curvature estimation for unstructured triangulations of surfaces. *Technical Report, Los Alamos National Laboratory* 2003.

[3] B. Hamann, Curvature approximation for triangulated surfaces. *In Geometric Modelling* (G. Farin at al., Eds.) Springer Verlag 1993, 139–153.

[4] E. Hameiri, I. Shimshoni, Estimating the principal curvature and the Darboux frame from real 3D range data. *IEEE Trans. Sys. Man. and Cyber.* **33** (2003), No.4, 626–637.

[5] P. S. Heckbert, M. Garland, Optimal triangulation and quadric-based surface simplification. *Computational geometry* **14** (1999), No.1, 49-65.

[6] Y.-B. Jia, L. Mi, J. Tian, Surface patch reconstruction via curve sampling. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, ICRA 2006.

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

[8] L. Linsen, K. Müller, P. Rosenthal, Splat-based ray tracing of point clouds. *Journal of WSCG* **15** (2007), 1–3.

[9] M.Meyer, M. Desbrun, P. Schröder, A.H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, (H.-C. Hedge, K. Polthier Eds.) Springer-Verlag 2002, 35–57.

[10] H. Pfister, M. Zwicker, J. van Baar, M. Gross, Surfels: surface elements as rendering primitives. In *SIGGRAPH'00*, ACM Press/Addison-Wesley Publishing Co., NewYork 2000, 335–342.

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

[12] S. Rusinkiewicz, Estimating curvatures and their derivatives on triangle meshes. In *2nd International Symposium on 3D Data Processing*, September 2004, 486–493.

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

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

[15] M. Szilvási-Nagy, Face-based estimations of curvatures on triangle meshes. *Journal for Geometry and Graphics* **12** (2008), No. 1, 63–73.

[16] M. Szilvási-Nagy, Construction of circular splats on analytic surfaces. in *Fifth Hungarian Conference on Computer Graphics and Geometry*, Budapest 2010, 55–57.

[17] G. Taubin, Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV 95)*, 1995, 852–857.

[18] S. Wolfram, *Mathematica. A System for Doing Mathematics by Computer. Second Edition*. Addison-Wesley, 1991.

[19] P. Yang, X. Qian, Direct computing of surface curvatures for point-set surfaces. In *Eurographics Symposium on Point-Based Graphics (2007)*, (M. Botsch, R. Pajarola Eds.), 2007.

Márta Szilvási-Nagy

Dept. of Geometry, Budapest University of Technology and Economics

H-1521 Budapest, Hungary

e-mail: szilvasi@math.bme.hu