3D POLYGONAL MODEL CURVATURE APPROXIMATION BASED ON LOCAL PROJECTIONS METHOD

Rostislav Hulík

Doctoral Degree Programme (2), FIT BUT

E-mail: xhulik00@stud.fit.vutbr.cz

Supervised by: Přemysl Kršek

E-mail: krsek@fit.vutbr.cz

Abstract: In this paper, a novel method for curvature approximation on 3D polygonal models is presented. The algorithm is based on local projections method, which represents a new way for information extraction from polygonal models. Curvature approximation is one of its possible applications, which has been analyzed and compared with today's common algorithms.

Keywords: EEICT, polygonal model, curvature, approximation, local projections

1. INTRODUCTION

Approximation of curvature on mesh surface is well known technique for describing the model's shape. The problem of curvature estimation on 3D polygonal mesh is not however as simple as computing curvature on analytical surface. The algorithm must deal with several factors resulting from polygonal representation, such as polygon inhomogenities or surface approximation (real curvature on each polygon equals zero). In this paper, a novel approach to the curvature approximation is presented, based on local projections of mesh surface. This method is primarily designed to overcome the problems stated above.

Local projections itself is also a new way to polygon model analysis which is necessary to describe. It is meant to simplify operations on polygonal meshes by converting the problem into well documented 2D image analysis. By projecting a mesh surface on a plane tangent to each vertex (with given size and resolution), a raster can be created. Then, a mesh can be understood as simple set of 2D matrices. With this easy modification, multiple computations common with image processing can be applied, e.g. curvature estimation, edge and blob detections, template matching or feature extraction methods such as SIFT and SURF.

As an evaluation tool, a curvature approximation using this method was developed and evaluated in this paper.

2. STATE OF THE ART

This paper pursues a general polygonal surface problem – approximation of curvature and its directions. Multiple algorithms are known for solving this issue, on following lines, several of them are briefly presented (This publication is not meant as an algorithm overview, so for precise information, please see individual references).

The problem of curvature approximation was well documented by Meyer, Desbrun, Schröder and Barr in [1], where discrete differential operators were presented. Curvature is computed directly from polygonal mesh structure by specifying discrete differential operators using angles around each vertex. The computation is normalized by area associated with each triangle (most common is Voronoi cell area). By this approach, Mean, Gauss, even minimal and maximal curvature can be approximated as well as principal curvature directions. Similar idea for curvature estimation – approximation directly from mesh structure – is described also by Rusinkiewicz [2].

Another method is presented by Simari, Singh and Pedersen in [3]. Their method uses a regularly resampled polylines (spider) with a center at each vertex. Spider is then used as a base for all consequent approximations. This algorithm apparently neutralizes main source of errors of mesh curvature approximation – irregularity of polygons.

There exist several algorithms which deal with curvature computation in a different way, for example by algebraic fitting of spheres onto the model (or point cloud in original paper by Guennebaud and coworkers [4]).

3. LOCAL PROJECTIONS METHOD

Local projections method is a novel approach to mesh processing. It is based on mesh rasterization on local tangent matrix, which converts a problem from 3D polygonal mesh processing task into 2D image processing.

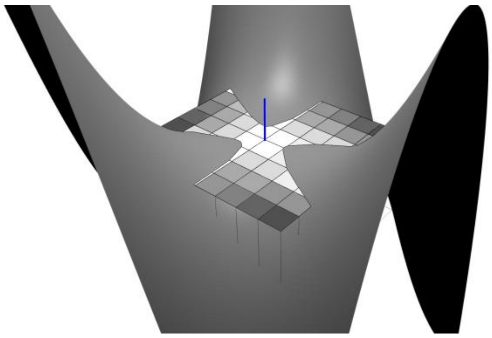

Figure 1. Tangent raster orthogonal projection of z-distance

Rasterization algorithm itself must be prone to holes in mesh, unconnected polygons and other options influencing consequent filtering and computations with conserving minimal time complexity. All requirements must be met with maintaining computation speed for usability of the whole procedure.

In order to fulfill all these requirements, a hybrid projection algorithm is proposed – a combination of polygon rasterization algorithm and ray-casting search. The combination was chosen to speed up a standard ray-casting search.

Algorithm for each vertex:

1. <u>Get recursively all vertices projected on matrix (neighborhood search) and rasterize them</u>

   This step was chosen for its relative speed - it searches recursively for all vertices which can be projected to tangent matrix. This is done because of minimum time complexity of vertex projection test. All polygons connected to them are rasterized.

2. <u>Rasterize triangles sharing rasterized boundary edges</u>

   Second point rasterizes all polygons connected to boundary edges. It is necessary for increasing number of filled pixels on tangent rasters. In Table 1, a histogram with the numbers of rasterized pixels in each step can be seen.

3. <u>Cast rays for all uninitialized pixels</u>

   Last step of the algorithm uses ray-casting speeded by 3D R-tree [6] and finishes the search. This step is the most time consuming, but increases the precision of computation. Tests showed that for precise use of this whole method, even one bad projected pixel can have great impact for future processing.

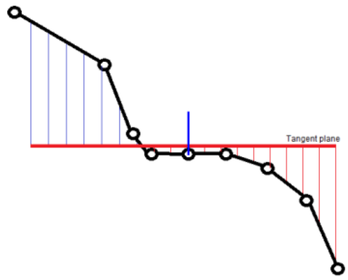

Figure 2. Z distance projection on tangent plane

4. CURVATURE APPROXIMATION

Rasters computed by proposed algorithm can be obviously used in various calculations. Curvature approximation is one of these possible applications.

Main goal of the curvature approximation is to approximate curvature behavior on analytical model. Since the polygonal mesh is the approximation of smooth surface, the curvature cannot be computed directly. By application of Hessian matrix, we can approximate minimal and maximal curvature along with principal curvature directions.

The Hessian H is the Jacobian matrix of second order partial derivatives of a function $f(x_1, x_2, \dots, x_n)$ with respect to x_1, x_2, …, x_n. In our context, given the image as a discrete function $I(i)$, we can obtain hessian as:

$$H(i,\sigma) = \begin{bmatrix} L_{xx}(i,\sigma) & L_{xy}(i,\sigma) \\ L_{xy}(i,\sigma) & L_{yy}(i,\sigma) \end{bmatrix} \quad (1)$$

where σ is the derivation neighborhood and i represents a specific point in image [5]. The derivative approximation L_x can be computed by convolution with normalized second order Sobel operators $g_{x_j}(\sigma)$ with specified size σ:

$$L_{xy}(i,\sigma) = \sqrt{\left(g_x(\sigma) * I(i)\right)^2 + \left(g_y(\sigma) * I(i)\right)^2} \quad (2)$$

Taking the $H(i,\sigma)$ of rasterized image with projected z-directions, we can simply approximate the curvature of a surface by eigen analysis on the Hessian. Eigen vectors will be equal to principal curvature directions and eigen values equal to principal curvatures respectively.

By this technique, we obtain a robust method for curvature approximation even on noisy model, because it is possible to apply one of smoothing kernels (gaussian, median) right onto tangent plane just before applying a Sobel operator. The smoothing has no effect on mesh topology, but has great effect on filtering of unwanted noise. Also, there is possibility to project a larger vertex neighborhood, which also cancels a significant amount of noise on a model (although this is against the correct definition of curvature, this action has great advantage on high noise models, such as marching-cubes created objects).

The projection from 3D to 2D can bring also several inaccuracies in curvature approximation. One of problems is small raster resolution, which can severely influence the output due to exceedingly coarse representation of vertex neighborhood. In this case, we can observe e.g. significant deflection of curvature directions from analytically computed curvature – the resolution 7x7 and larger is recommended.

Another problem can result from deeply curved mesh, which will be cropped by simple projection into 2D. In this case, projected surface will be different from the original (real surface can be “behind” already rasterized pixel). This problem can be however evaded by using small matrix size. In real data sets, this phenomenon is negligible.

5. METHOD EVALUATION

5.1. RASTERIZATION ALGORITHM

On preceding lines, an algorithm for fast and reliable software mesh rasterization on tangent matrices was presented. For rasterization itself, a hybrid algorithm was chosen, due to intention to speed up the rasterization process.

Tests have shown that simple ray-casting step can find all of possible projections, but this approach is too slow to be used in software implementation. As tests have proven on testing dataset (data consisting various set of polygonal models), 91.59% of pixels in average are found by step 1 (see chapter 3). After step 2, 95.08% (average) of pixels are found on tangent plane. Ray casting step completes 98.87% (average) of pixels – the non 100% efficiency is due to pixels which are projecting infinity (there is no mesh to project on z-distance).

Use of hybrid algorithm speeded up the computation of meshes at least 10 times comparing to simple ray-casting algorithm. See Table 1 for comparison of time consumption (size of matrix is relative to model – it means median of edge lengths multiplied by **size**). The time of the whole computation is dependent not only on the mesh size, but also on the mesh topology (number of vertices searched by the Ray casting).

	Matrix resolution 3x3, Size 2		Matrix resolution 9x9, Size 9		Curvature Schröder et al. [1]	
No of vertices	**Average (ms)**	**Median(ms)**	**Average(ms)**	**Median(ms)**	**Average(ms)**	**Median(ms)**
~20000	1478,71	1358,41	16546,2	14821,2	238,9	237,62
~30000	2088,82	1674,32	20328,55	14824,65	372,94	377,02
~50000	4433,09	2782,42	36448,18	18381,6	619,71	620,15
~200000	17714,03	12355,1	101518,96	93521,2	2538,65	2546,08

Table 1: Speed of computation on various meshes (Intel Core 2 Duo, 2.53 GHz)

Results are compared with speed of curvature computation of Schröder and his coworkers [1]. The speed of curvature computation (it takes into account direct vertex neighborhood) is comparable with rasterization of 3x3 neighborhood with size equal to 2. Implemented software projection is clearly slower than compared processing method, but it can be further accelerated by parallelization. Also, hardware implementation using GPU is suggested to introduce a significant speed up of computations.

5.2. CURVATURE APROXIMATION

Curvature approximation method was tested by comparison of approximated curvature on triangulated analytical surfaces with analytically computed curvature, Schröder's method and algebraic point set surfaces method (APSS) described above [4]. Analytical method was computed from mathematically defined surfaces. These models were triangulated (intentionally irregularly) and were used as inputs for tested methods for comparison. During comparison, method outputs were normalized to achieve best comparison results. In Figures 3 and 4, test outputs can be seen from two different analytical surfaces (horizontal axis equals to vertex indices, vertical to normalized curvature approximation).

It is clear that Schröder's method along with APSS are more precise in approximation of well triangulated mesh, however it has difficulties when noise or triangulation irregularities or unconnected vertices in mesh are present (see Figure 3 around vertex no. 41). On irregularly triangulated mesh, our method has very similar behavior as analytical computation.

On Figure 4, it is clear that the major advantage of proposed algorithm is its insensibility to noise. To the analytical mesh, randomly generated noise was added. Even the APSS holds major extrema due to variable fitting radius, perturbations are still present on low curvature regions. This fact can lead to possible application of this approach to tasks of noised meshes processing and analysis.

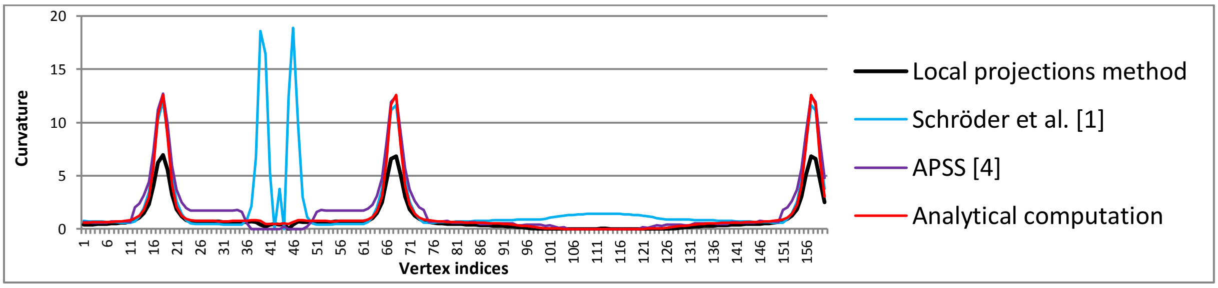

Figure 3. Surface $z = \cos(x^2 + y^2)$ – maximum curvature

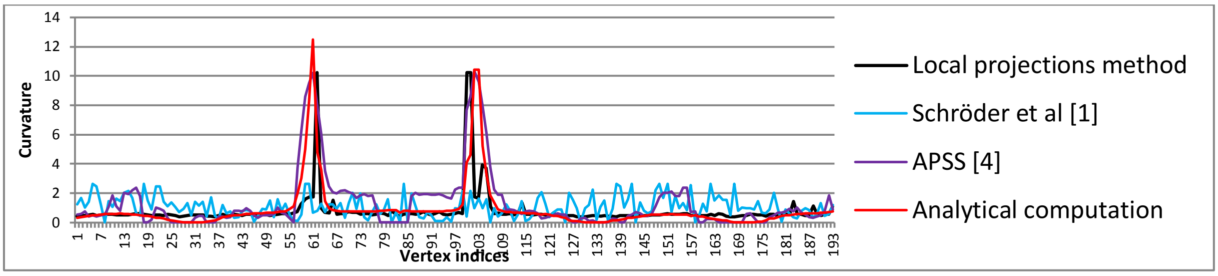

Figure 4. Surface $z = \cos(x^2 + y^2)$ – maximum curvature, added random noise to triangulated model

6. CONCLUSION

A novel approach to the mesh evaluation and analysis was presented. Local projections method is a widely applicable tool for mesh analysis which allows using arbitrary 2D image processing operator directly on mesh structure extending mesh processing with significant number of new algorithms. Additionally, this method is capable of smoothing and polygon irregularities cancelling without any mesh topology change.

Also, following the idea of local projections method, a novel way for mesh curvature approximation was created. Even there exist more precise algorithms for curvature approximation on polygonal meshes, our approach has great advantage to be prone to unwanted noise on mesh, easily controllable by adjusting local projections matrix size.

ACKOWLEDGEMENT

The research leading to these results has received funding from research project MSM0021630528 - Research information technology security (2007-2013, MSM).

REFERENCES

[1] Meyer, M., Desbrun, M., Schröder, P., Barr, A. H.: *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Visualization and Mathematics III (p. 35-57), Springer-Verlag, Heidelberg, 2003

[2] Rusinkiewicz S: *Estimating Curvatures and Their Derivatives on Triangle Meshes*. In Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium (3DPVT '04). IEEE Computer Society, Washington, DC, USA, 486-493, 2004

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

[4] Gaël Guennebaud and Markus Gross. 2007. Algebraic point set surfaces. ACM Trans. Graph. 26, 3, Article 23, July 2007

[5] K. Mikolajczyk and C. Schmid. 2002. *An Affine Invariant Interest Point Detector*. In Proceedings of the 7th European Conference on Computer Vision-Part I (ECCV '02), Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (Eds.). Springer-Verlag, London, UK, UK, p. 128-142, 2002

[6] Guttman, A.: *R-Trees: A Dynamic Index Structure for Spatial Searching*. Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84. pp. 47, 1984