Local Projections Method and Curvature Approximation of 3D Polygonal Models

Rostislav Hulík
Faculty of Information Technology, Brno University of Technology
Božetěchova 1/2
612 66, Brno, Czech Republic
ihulik@fit.vutbr.cz

Přemysl Kršek
Faculty of Information Technology, Brno University of Technology
Božetěchova 1/2
612 66, Brno, Czech Republic
krsek@fit.vutbr.cz

ABSTRACT

Mesh processing is a wide area which consists of several approaches, heavily specific for each task. We present a novel approach to mesh processing using the Local Projections method. The described method makes benefit of wide variety of image processing algorithms, very similar to 3D tasks, and implements a conversion mechanism to make possible use of these processes on polygonal models. We also designed, implemented and evaluated the curvature approximation algorithm to test and compare the proposed method usability on real and artificial data. Use of the proposed method brings significant benefits especially to noised mesh analysis.

Keywords

Local Projections, Curvature, Polygonal model, Mesh Processing, Image Processing

1. INTRODUCTION

Polygonal model processing techniques consist of a wide variety of approaches specific for each different task. Each of these methods is especially designed for its purpose, e.g. curvature approximation, mesh smoothing, simplification, matching or feature extraction. Such diversity leads to the separation of this field of study into several different domains.

Even the problems stated above are handled separately, they need to overcome the same problems emerging from the polygonal mesh data structure itself. This includes irregularities of polygons, unconnected parts of the mesh or the polygonal surface approximation itself.

In the computer graphics area, another widely used and well documented area exists with very similar operations. The image processing scope consists of well-known techniques for curvature approximation on raster images, smoothing kernels or feature extraction methods. The operations are very similar to these used in polygonal mesh processing.

The main problem appears from different data structures. We present a consistent and robust way of polygonal model representation, where each vertex is described by its own tangent raster depicting its neighborhood. In this way, we can apply arbitrary raster operators right onto the mesh and extract the necessary information. We are able to approximate the curvature, extract specific features or apply other operators, such as smoothing, right on the tangent rasters resulting in smoothing for successive operations. All of these operations can be done only on tangent rasters without alternating the original mesh topology.

As a usability demonstration, we have chosen implementation of curvature approximation, which can be easily evaluated by comparing not only with existing methods, but also with an analytical approach, which allows us to demonstrate the accuracy of our method. In the evaluation chapter, we demonstrate that this method brings significant advantages to the smoothing of largely noised meshes.

In the first part of this paper, we describe deeply the Local Projections (LP hereafter) method to understand fully the proposed conversion. The second part is concerning with curvature approximation methods using the proposed LP method. In the concluding part, we present test results gathering the efficiency of LP and the accuracy of curvature approximation using this approach by comparison with today's commonly used methods and analytically computed curvature. Comparison is done on both smoothed and noisy data sets. Last section is devoted to conclusions, future goals and possible improvements of LP.

2. RELATED WORK

As we stated above, the mesh processing field of study is a wide area consisting of many different approaches. In this section, we describe today's methods for curvature estimation on mesh structures in order to compare them with our LP curvature approximation.

The most widely known algorithms for mesh curvature estimation are the discrete differential operators presented by Meyer et al. [Mey00a]. The authors describe operators for direct approximation of curvature from mesh structures considering angles of adjacent edges. The operators exist for both Gauss and mean curvature and it is possible to induce minimal and maximal curvature respectively. Although the computation is normalized by the Voronoi area of neighboring faces, the method fails when any triangle irregularities or unconnected faces occurs. On the other hand, the approximation is very straightforward which significantly influences the efficiency. A similar method of computing directly from mesh structure is described by Rusinkiewicz [Rus00a].

Another approach makes benefits from geometric fitting of spheres directly onto mesh structures. The algebraic point set surfaces method (APSS) was presented by Guennebaud and Gross [Gue00a] [Gue00b] as an algorithm for point cloud curvature estimation, but it was also adapted to point clouds with normals and polygon meshes. Due to the approximation of surface with spheres, triangular irregularity problem is overcome. The algorithm also partially solves the problem of unconnected polygons (we fit spheres with certain radii – the time complexity increases) and the mesh holds its curvature features even on noisy data.

Simari et al. [Sim00a] presented a method for robust curvature estimation, which is based on regularly resampled polylines with intersections in current vertices and specified angular steps. From these resampled polylines (forming spider shape around current vertex), authors simply compute curvatures in specified directions given by each branch and imply principal curvature values and directions. It is clear that the authors also try to overcome problems of polygon mesh irregularities using resampling procedures. Unconnected polygons are not dealt with in the work presented.

Page et al. [Pag00a] proposed another method aimed especially at noisy datasets. Their approach tries to find a geodesic neighborhood of a vertex with specified distance, which gives us the possibility of smoothing high frequency noise. It is necessary to underline the fact that the authors do not use an Euclidean distance to estimate geodesic neighborhoods, but the shortest geodesic path. Selected vertices then vote to Taubin's curvature tensor [Tau00a] from which curvature is estimated. This method has proven to be very robust against the noise of meshes and due to possible resampling of vertices on geodesic neighborhood, robust against irregular triangulation.

There exist several other approaches to curvature estimation, described e.g. in [Ozt00a][Mok00a]. The main traits of novel methods are always the same – to overcome mesh tessellation irregularities, unconnected elements, noise and polygonal approximation. It is also clear from these examples of curvature processing that each method is developed especially for its purpose, as stated in the introduction. We tried to approach this problem from the other, well documented field – image processing.

3. LOCAL PROJECTIONS METHOD

In this section, a method for the 3D – 2D problem conversion is presented. The main purpose is to create a novel approach to mesh processing by turning the problem of polygonal mesh analysis to an raster image problem.

For vertex neighborhood representation in 2D image rasters, we have chosen to project distances from the tangent plane at a given vertex position. This operation results in a rasterized matrix of z-distances (depth image) attached to each vertex. However, it is possible to project arbitrary vertex information, e.g. color, curvature or other vertex specific values present in the mesh structure. In the following lines, we consider always depth image representation.

Rasterization must overcome all of the problems stated above, such as unconnected polygons, holes in meshes etc. All of this must be done while conserving minimal time complexity to maintain usability of the whole procedure.

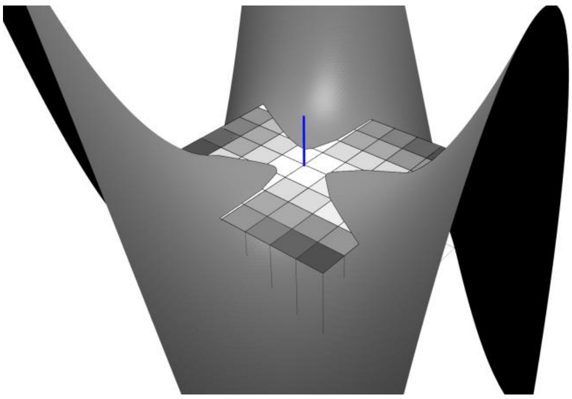

Figure 1. Tangent raster with projected z-distance

In Figure 1, an example of such a projection is presented. Tangent rasters are defined **for each vertex** and describe the vertex neighborhood.

Regarding this fact, size of matrices must be chosen appropriately to the specific task. For example when using LP method for curvature approximation, we choose smaller matrix size due to the definition of curvature – we expect the analysis of small neighborhood around pixel. On the other hand, when using the LP method for i.e. feature extraction on mesh structures points of interest, we can describe vertices with matrices of larger size and resolution.

Tangent plane direction itself is computed for each vertex using its normal vector defined as:

$$\overrightarrow{n(x)} = \frac{\sum_{i \in F(x)} \overrightarrow{n(i)}}{|F(x)|} \quad (1)$$

where x is current vertex and $F(x)$ is a set of neighboring faces. $F(x)$ consists of not essentially adjacent faces to the current vertex – if larger smoothing is required, F can contain larger neighborhood.

In the following sections, we mention two parameters of matrices. The **resolution** stands for matrix size in pixels (matrix is square) and **size** is the relative size of the tangent plane to the median of model edge lengths. E.g., if the median of edge lengths is equal to **x** in model coordinates and the matrix size is set to **s**, the real edge size of the matrix will be equal to **xs.**

Matrix resolution and size is highly specific to the task we request from LP method. If LP is used for e.g. edge detection or curvature approximation, it is necessary to apply relatively small resolution (Sobel operators in image processing hardly exceed 9x9 resolution) and small size of matrix (the larger matrix is, the larger smoothing occurs). On the other hand, SIFT descriptors need large pixel neighborhood to compare, so size and resolution can be much higher.

In subsequent computations, each vertex matrix is treated individually – we can apply an arbitrary image processing operator to every vertex such as curvature computation, edge/blob detector, feature extractor etc. The value of raster's center pixel is the value of the used operator for corresponding vertex.

Rasterization algorithm analysis

To achieve the exact projection on tangent plane multiple algorithms can be used. We consider three possibilities:

1. **Blind rasterization of the whole mesh on each raster**

   This method rasterizes precisely even meshes with unconnected polygons and holes. It has, however, a high time consumption. On the other side, it is well parallelizable and we can imagine this solution in possible future hardware implementation.

2. **Rasterization of vertex neighborhood**

   By rasterizing only a vertex neighborhood, we can achieve a very fast projection procedure. This approach works only for well-connected meshes and can pose problems when the holes occur in the proximity of the current vertex.

3. **Ray-casting rasterization**

   Ray-casting of each matrix cell also accomplishes a total projection without problems. It has, however, the same problems as possibility 1 – an extreme time complexity.

In order to fulfill all quality requirements of projection while maintaining time complexity as low as possible, a hybrid algorithm is proposed combining all three steps described above.

In Figure 2, the final proposed algorithm overview is presented.

```
For each vertex:
  I.    Get all neighboring vertices
 II.    Rasterize triangles connected
        to them
III.    Rasterize borders
 IV.    Ray-cast untouched pixels
```

Figure 2. Rasterization algorithm

In **step I**, all neighboring vertices are found, which will be projected onto tangent raster. The maximum distance must be specified to prevent passing the whole mesh in the case when faces are perpendicular to the projection plane (see Figure 3). This step is followed by **step II** which is very fast and poses minimal computing and efficiency problems.

Step III blindly rasterizes the borders of already found segments. This apparently useless step has the purpose of finding triangles which have all vertices outside projection plane (Figure 3 – yellow segments). Steps I-III can clearly be replaced by traversing triangles instead of vertices but this approach will increase extensively the time complexity – each triangle needs to be tested for possibility to project it onto the tangent plane - in our approach, we test only point projection.

The blind border rasterization cannot, however, find all possible interfering polygons (e.g. border search rasterizes only one-neighborhood). In **step IV**, we ray-cast the rest of provisionally non- rasterized pixel to assure the certainty of all pixels rasterization. The ray-casting can be further accelerated by using some high-level search structure such as 3D R-Tree [Gut00a] etc.

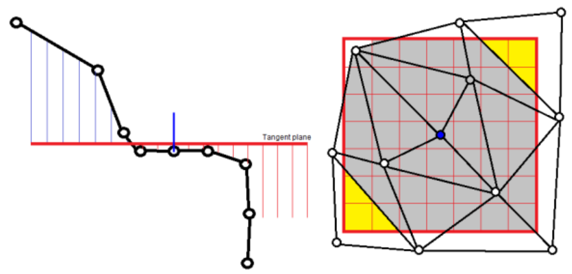

Figure 3. Problems in rasterization, infinity projection (left) and triangles having all vertices outside projection matrix (right)

All of the four rasterization steps were chosen for maximum speed-up of the whole procedure. In the evaluation section, each of the steps is evaluated and diagnosed to provide the reason for necessity of each step.

4. CURVATURE APPROXIMATION

The rasters computed by our algorithm can be used for arbitrary feature extraction of the polygon mesh. We have chosen curvature extraction for the reason that it can be easily compared with the analytical approach, i.e. we can prove the feature conservation by conversion from a polygonal mesh to an image problem.

The main goal of curvature extraction is to approximate the curvature behavior of an original model. It is necessary to accentuate that the real curvature on the polygon mesh cannot be taken into account because of null curvature on mesh faces and infinite curvature on edges.

To approximate the curvature and its directions from depth map (i.e. our tangent rasters), we can apply the Hessian matrix analysis to approximate the curvature behavior. For continuous functions, the Hessian H is the Jacobian matrix of derivatives $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}$ of a function $f(x_1, x_2, \ldots, x_n)$ with respect to x_1, x_2, …, x_n:

$$Hf(x_1, x_2, \ldots, x_n) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{vmatrix} \quad (2)$$

The Hessian H is a simple matrix of partial second order derivatives of a continuous function. In our case of a discrete raster, we must define a second derivative operator to approximate the derivative behavior on raster image. For this purpose, we have chosen second order Sobel operators to compute a derivative estimation L_{xy} by convolution with the input raster (equation 3). It is clear that the convolution is not necessary to be executed for all raster pixels, but only for the center, i.e. for the current vertex.

$$L_{xy}(i,\sigma) = \sqrt{\left(g_x(\sigma) * I(i)\right)^2 + \left(g_y(\sigma) * I(i)\right)^2} \quad (3)$$

Taking computed partial derivatives, we can easily construct a discrete equivalent of Hessian (equation 4).

$$H(i,\sigma) = \begin{bmatrix} L_{xx}(i,\sigma) & L_{xy}(i,\sigma) \\ L_{xy}(i,\sigma) & L_{yy}(i,\sigma) \end{bmatrix} \quad (4)$$

where σ is the derivation neighborhood (i.e. matrix size) and i represents a specific point in the image [Mik00a]. Corresponding curvatures and directions can be subsequently computed by eigen analysis of the computed Hessian (equation 5):

$$HX = \lambda X \quad (5)$$

Where λ is an eigenvalue and X the corresponding eigenvector. For a 2x2 Hessian matrix, we obtain precisely two eigenvalues and two eigenvectors corresponding to approximated curvatures on the tangent plane. The necessity of backward transformation to 3D coordinates is evident.

By this technique, we can obtain a robust curvature estimation technique even on noisy meshes due to the possibility of off-line smoothing right on the tangent rasters. Off-line smoothing in this case means the smoothing only on raster representation, without any mesh topology change. Such smoothing can be achieved by application of one of the image smoothing kernels, such as median, Gaussian or mean. Smoothing can be also achieved by application of some polynomial interpolation which can easily break hard mesh edges and approximate the real object surface.

Another noise cancelling feature of the LP method is the possibility of larger vertex neighborhood projection. Even this fact is against the formal curvature definition, it can be used on very noisy models when a larger neighborhood is necessary for real object feature extraction. The idea of a larger vertex neighborhood was also presented in [Gue00a] [Gue00b] [Sim00a] [Pag00a] as a means for noise cancelling. For LP, this feature produces another significant advantage – the polygon structure cancellation. By the rasterization of larger neighborhoods the image is normalized – each vertex is represented by a map of the same size.

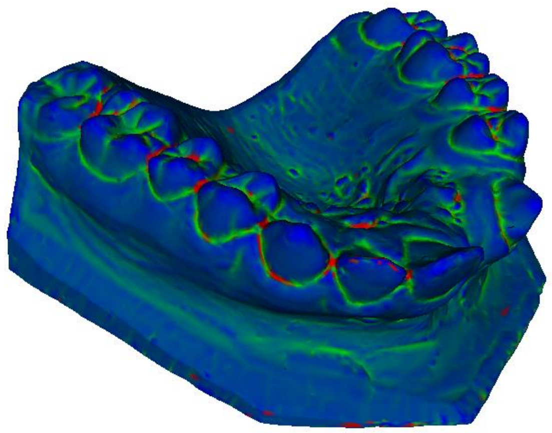

Figure 4. Maximum curvature computed from 7x7 size matrices of the size equal to the mean size of vertex one-neighborhood.

The LP curvature approximation method can also yield several drawbacks emerging from the rasterization and 3D-2D conversion.

The first produces problems when small tangent resolution is applied which can severely influence the output due to exceedingly coarse representation of vertex neighborhoods. In this case, we can observe e.g. significant deflection of curvature directions from analytically computed curvature. In the curvature computation case, it is recommended to use at least a 7x7 raster (This value was chosen experimentally as a compromise to the speed of rasterization process and accuracy of approximation). For other applications, such as SIFT or SURF descriptors, much greater resolution can be required.

Another problem can result from highly curved meshes. The projection of these meshes can result in cropping of the real surface in projection into 2D (real surface can lie "behind" already rasterized pixels – see Figure 5, yellow section). Although this fact can lead to very imprecise representation of the original mesh shape, the tests on real datasets show that this phenomenon is extremely rare due to the size of matrices (which rarely exceed one- or two-neighborhood). For curvature computation, we semi-occasionally use matrices of such large neighborhoods – in our computations, we rarely exceed one-neighborhood of corresponding vertex. On the other hand, this effect can be applied for noise canceling and detail smoothing for coarse feature extraction.

5. EVALUATION

We designed multiple tests for the proposed methods evaluation. The evaluation section is separated into two major sections – evaluation of the rasterization algorithm and evaluation of curvature approximation.

Local projections method

On preceding lines, a robust projection method of the mesh onto tangent raster was presented. As a first evaluation method, we compare the number of projected pixels at each step of rasterization.

	Average	**Median**	**Std. dev.**
After step 2	91,59%	95,08%	10,36%
After step 3	94,31%	98,25%	10,40%
After step 4	95,20%	98,87%	10,19%

Table 1. Pixels projected in each step

In Table 1, the number of projected pixels at each step is presented. We tested our approach on various polygonal meshes. In the testing dataset, there are closed meshes, data with holes and non-connected vertices and other problematic cases described above.

As tests have proven, a majority of pixels are filled in steps 1 and 2 (see the section Local Projection Method). These steps are the quickest part of the rasterization. Step 3 will rasterize up to 94,32% of pixels – it is clear that this modification plays an important role in efficiency improvement, because by simple blind rasterization we fill up a significant partition of unprocessed pixels. For ray-casting step 4, only 0,89% of pixels remain – this leads to minimal computing complexity – ray casting is used only if there is no possibility to find the polygon by neighbor search. The non 100% efficiency is due to the pixels which are projecting infinity – in the post-processing step, these are filled by a chosen maximum value – we propose this value to be equal to the size of the tangent matrix (which is the same for all vertices).

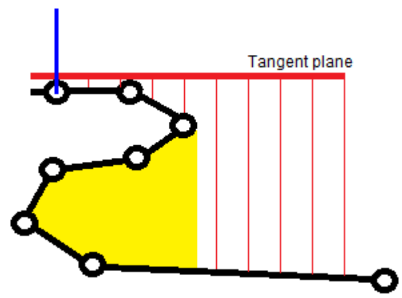

Figure 5. Problems on significantly curved mesh

	Matrix resolution 3x3, Size 2		Matrix resolution 9x9, Size 9		Curvature Meyer et al. [Mey00a]	
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 2. Speed efficiency of LP method compared with Meyer et al.

By this modification, we assign a specific, non-infinite value, to each uninitialized pixel. The matrix is then able of being processed by any image processing algorithms without creating infinity artifacts. We have chosen a "cube" maximum restriction due to the expected behavior of subsequent methods – a large, possible infinite value would cause a large perturbation in succeeding operator. In this case, LPM will lose its smoothing capabilities. We need then some value which represents a maximum possible value (size of tangent matrix) but does not cause such problems. The maximum z value can be however set as a parameter of computation. The right choice will surely influence the robustness of this method with respect to the mesh imperfections.

The second test is a speed comparison between our method and the well-known curvature computation method by curvature operators [Mey00a]. We have chosen this approach to compare with our method due to similar neighborhood search and feature extraction. This operation is comparable with rasterization of 3x3 matrices of size equal to 2, because it takes into account approximately the vertex neighborhood of the same size.

In Table 2, the time test results are presented. The LP method is clearly more time expensive than simple neighborhood search, which was expected. The effectiveness is dependent not only on matrix resolution or size, but also on mesh structure – on the number of pixels searched by ray-casting (mesh complexity), number of holes and inconsistencies.

Even the software implementation is significantly slower than simple neighborhood search, the efficiency is redeemed by the wide possibilities of this method. The method is largely parallelizable, so hardware implementation is expected to be prepared in the near future.

Curvature approximation

Curvature approximation is the second type of test provided in this paper. We prepared a testing dataset consisting of both real world objects (created by 3D scanning or conversion from volumetric datasets) and artificial data. Artificial data are primarily analytical shapes, where curvature it is possible to compute analytically. Both types of data were also used with added noise.

We compared our LP method with two common curvature approximations – the differential geometry operators of Meyer et al. and APSS (see Related work). Additionally, on analytical models, the curvature was also compared with analytical computations (AC hereafter).

In Figure 6, a comparison of the $z = \cos(x^2 + y^2)$ surface is presented. To the model, some unconnected vertices were artificially added to demonstrate the method's behavior in such case. Such unconnected polygons/vertices are common problem on polygonal meshes created by e.g. point cloud reconstruction. From this figure it is clear, that both Meyer's and the APSS methods are more precise in curvature approximation on well triangulated surface (visible peaks are well aligned with analytically computed curvature). The LP method holds the progression of AC.

A different testing schema occurs when artificial noise is added to the mesh (see Figure 7). The noise was added by the arbitrary fractal displacement method. In this case, the main advantages of the LP method are visible. Even the APSS holds major extrema due to the variable fitting radii, perturbations are still present on low curvature regions. The LP method holds all major progressions of the analytical computation. This fact makes this method very suitable for noised and incomplete meshes, where the curvature can be approximated very robustly.

6. CONCLUSION

A novel approach for mesh analysis has been presented along with one validation algorithm – a curvature approximation. The main contribution of this method is that it brings possible conversion of 3D polygonal mesh problem to the raster image processing issue. By this modification, we are able to apply multiple operators well known from image processing area right onto the mesh structure and treat it like simple raster image.

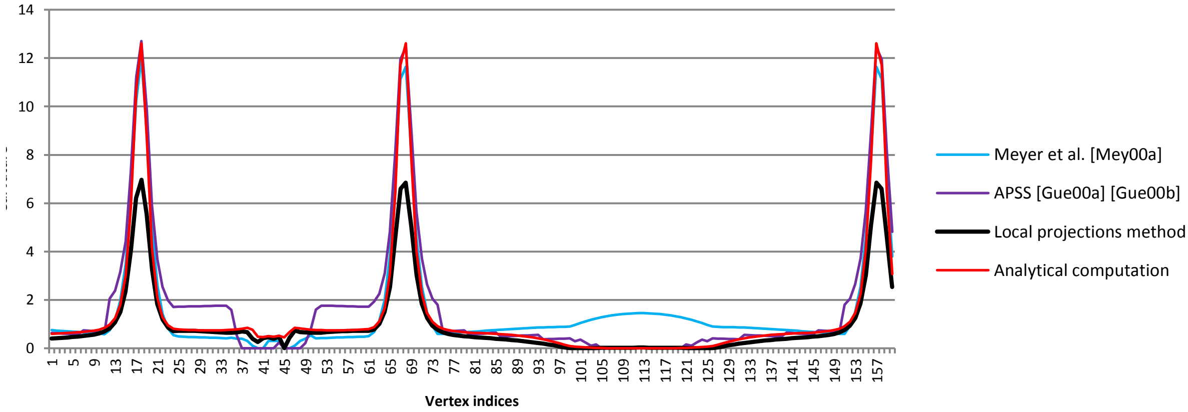

Figure 6. Surface $z = \cos(x^2 + y^2)$ – maximum curvature

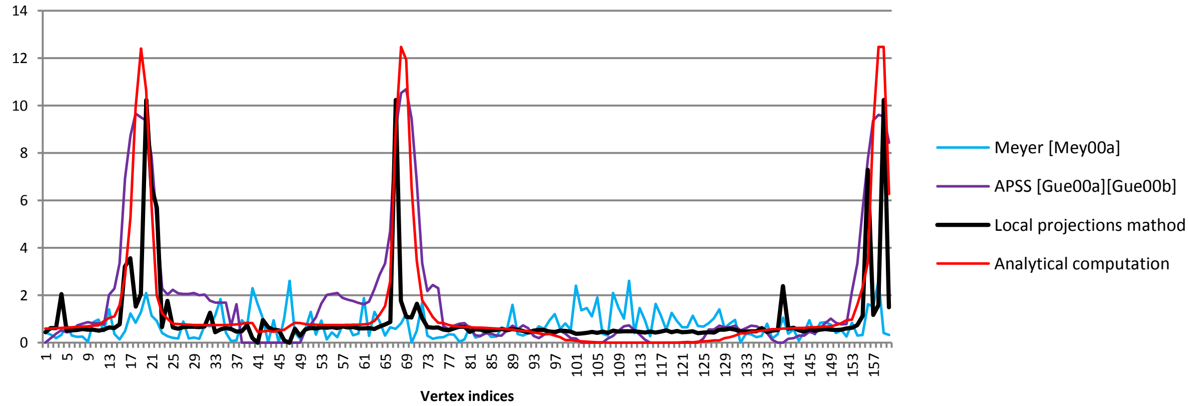

Figure 7. Surface $z = \cos(x^2 + y^2)$ – maximum curvature, added random noise to triangulated model

As tests have shown, the LP method has a big disadvantage in efficiency because of multiple projections computed for each vertex of the mesh. These problems were partly solved by application of a robust rasterization algorithm which speeded up the projection process. Additionally, due to the proposed hybrid method, we are able to analyze an arbitrary mesh – even meshes with topological inconsistencies, holes, unconnected elements, noise etc. Due to the method's simplicity, a hardware implementation is possible which is supposed to significantly speed up the computation of rasters.

The main advantage of the LP method is that it allows us to apply an arbitrary image processing operator right on the 3D mesh structure, which is in our case represented as a set of vertex tangent rasters. It is then possible to apply e.g. curvature approximations or feature extractors from the well-known image processing area. In preceding articles, multiple drawbacks associated with 3D to 2D projection were discussed. In tests on real and artificial data, these projection problems are irrelevant and they influence the subsequent computation minimally.

We also presented a new curvature approximation method, which comes from the LP method and image processing. The curvature is computed using Hessian matrix analysis from arbitrary sized tangent matrices, which allows us (in combination with smoothing operators) to estimate curvatures on any mesh even with structural noise present. The smoothing and noise cancelling capabilities of method presented makes the LP method a perfect candidate in the field of feature extraction from damaged and noised meshes (Figure 8).

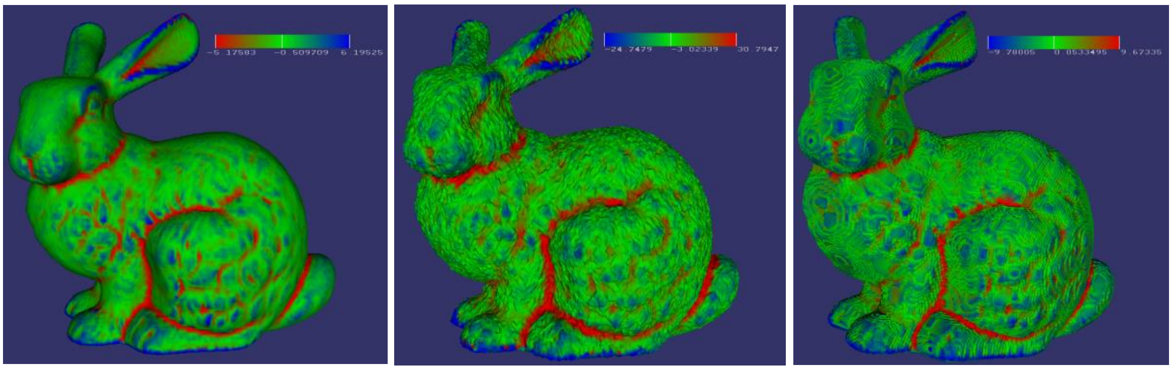

Figure 8. Mean curvature comparison on smoothed mesh (left), mesh with added random noise (middle) and marching cubes created model (right)

Future work in this field of study can be aimed at already mentioned hardware implementation of the LP method, which can significantly increase the computation speed of all methods and make our method comparable in efficiency with other largely used vertex-neighbor searching algorithms. Another research tendency will aim at other image processing operators possibly usable with LP method, especially in the field of mesh matching and feature extraction.

7. ACKNOWLEDGMENTS

This work was supported by the European Community's 7th Framework Artemis JU grant agreement n. 100233 (R3-COP) and European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

8. REFERENCES

[Mey00a] 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

[Gue00a] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces: ACM Trans. Graph. 26, 3, Article 23, July 2007

[Gue00b] G. Guennebaud, M. Germann, and M.H. Gross, Dynamic Sampling and Rendering of Algebraic Point Set Surfaces. In Proceedings of Comput. Graph. Forum. 2008

[Gut00a] 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

[Mik00a] Mikolajczyk, K., Schmid, C.: 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

[Mok00a] Mokhtarian, F., Khalili, N., Yuen, P.: Curvature Computation on Free-Form 3-D Meshes at Multiple Scales, Computer Vision and Image Understanding, Volume 83, Issue 2, August 2001

[Ozt00a] Oztireli, C., Guennebaud, G. and Gross, M.: Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression: In proceedings of Computer Graphics Forum 2, vol. 28, 2009

[Pag00a] D. L. Page, Y. Sun, A. F. Koschan, J. Paik, and M. A. Abidi. 2002. Normal vector voting: crease detection and curvature estimation on large, noisy meshes. Graph. Models 64, 3-4 (May 2002)

[Rus00a] 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.

[Sim00a] 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

[Tau00a] G. Taubin, Estimating the tensor of curvature of a surface from a polyhedral approximation, in Proceedings of the Fifth International Conference on Computer Vision, 1995, pp. 902–907