

A Roughness Measure for 3D Mesh Visual Masking

Guillaume Lavoué*
LIRIS UMR 5205 CNRS
INSA-Lyon, F-69621, France

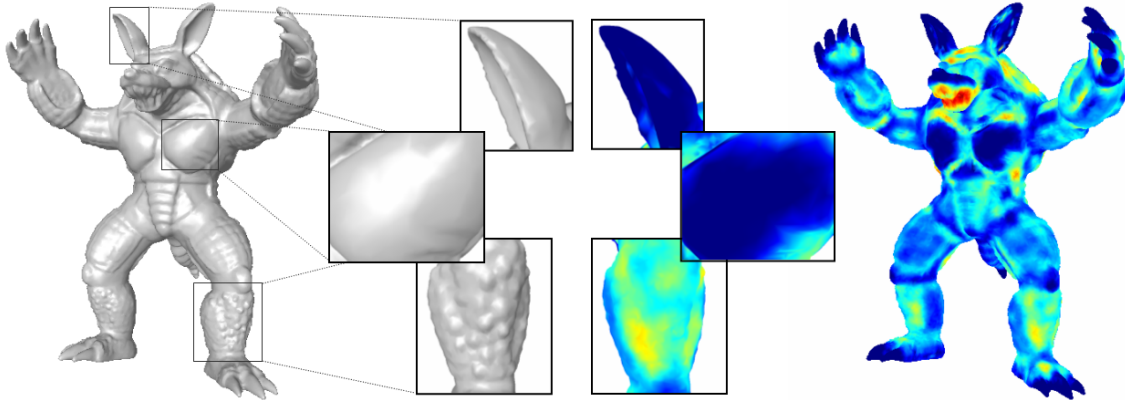


Figure 1: The Armadillo 3D model and its *roughness* map. Warmer colors (reds, yellows, greens) illustrate high roughness values while cooler colors (dark blue) illustrate rather smooth regions.

Abstract

3D models are subject to a wide variety of processing operations such as compression, simplification or watermarking, which introduce slight geometric modifications on the shape. The main issue is to maximize the compression/simplification ratio or the watermark strength while minimizing these visual degradations. However few algorithms exploit the human visual system to *hide* these degradations, while perceptual attributes could be quite relevant for this task. Particularly, the Masking Effect defines the fact that a signal can be masked by the presence of another signal with similar frequency or orientation. In this context we introduce the notion of *roughness* for a 3D mesh, as a local measure of geometric noise on the surface. Indeed, a textured (or *rough*) region is able to hide geometric distortions much better than a smooth one. Our measure is based on curvature analysis on local windows of the mesh and is independent of the resolution/connectivity of the object. An application to Visual Masking is presented and discussed.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.m [Miscellaneous]: Perception;

Keywords: 3D mesh, Roughness, Curvature, masking.

1 Introduction

Technological advances in the fields of telecommunication, computer graphics and multimedia during the last decade, have contributed to an evolution of the digital data being manipulated,

visualized and transmitted over the Internet. Nowadays, three-dimensional data (mostly represented by polygonal meshes) constitute the emerging multimedia content. In this context, 3D models are subject to a wide variety of processing operations such as compression, simplification, indexing or watermarking, which can introduce degradations of the visual quality of the shape. These processes are generally committed for human centred applications, thus it seems relevant to incorporate human perception principles in their realization.

Many computational models of the human visual system have been developed in the field of image processing particularly to evaluate the perceptual similarity between two images like the Visible Difference Predictor (*VDP*) [Daly 1993]. These models often rely on some psychophysical and physiological evidences like the *Masking Effect* which defines the fact that a signal can be masked by the presence of another signal with similar frequency or orientation.

In the field of computer graphics such perceptual models have not been really considered except for rendering: Perceptual metrics are used to determine, according to the location of the observer, which Level Of Details (*LOD*) to use to satisfy frame rate and image quality requirements [Luebke and Hallen 2001; Reddy 2001]. Little attention has been paid to the use of perceptual models for standard 3D mesh processing. Some *ad-hoc* metrics have been proposed to pilot mesh simplification [Kim et al. 2002], evaluate compression [Karni and Gotsman 2000] or watermarking algorithms [Rondao-Alface et al. 2005; Drelié Gelasca et al. 2005; Corsini et al. 2005] or predict the visual distortion between two meshes [Lavoué et al. 2006]. However, except the later, these existing metrics are designed to measure specific artifacts produced by specific applications. Few general-purpose measures have been developed that would be able to guide or improve any kind of processing (simplification, compression, watermarking, recognition etc.). One exception is the work of Lee et al. [Lee et al. 2005] which aim at detecting perceptually salient regions of a 3D object, using some curvature statistics. Their objective is to drive some processing operations, like simplification for instance, in such a way so as to preserve these visually salient features. Similarly Howlett et al. [Howlett et al. 2005] determine the saliency through an eye tracking system.

*e-mail: glavoue@liris.cnrs.fr

Our objective is somehow dual: We want to exploit the human visual system to *hide* geometric degradations, produced by standard operations. The key idea is to detect regions where slight geometric distortions would be near invisible. This idea is linked with the concept of Visual Masking: A *rough* (or *noised*) region is able to hide some geometric distortions, provided that their frequencies are quite similar. For instance, in figure 2, the artifacts produced by a geometric quantization (8 bits) are much more visible on the smooth surface (left) than on the rough surface (right). Indeed the visual texture produced by roughness has masked the geometric modifications. In the field of Computer Graphics, this masking effect was only investigated by Ferwerda et al. [Ferwerda et al. 1997]. They propose a computational masking model, extending the Daly VDP, which illustrates how a texture map can mask the polygonal tessellation. Our objective is not to propose such complex computational masking model but rather a simple roughness estimator allowing to concentrate geometric errors coming from common processing operations on areas associated with high masking properties. This concept of *roughness* is quite relevant for 3D

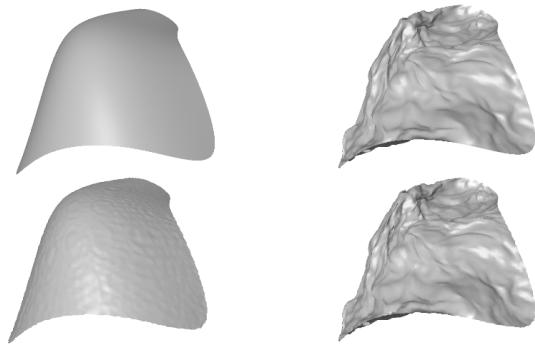


Figure 2: Effect of roughness for 3D masking. The upper images are original while lower ones are quantized on 8 bits. The quantization artifacts are much more visible on the smooth surface than on the rough one because of the masking effect.

perception. A lot of roughness-like measures have been considered in most of existing perceptual 3D metrics. Karni and Gotsman [Karni and Gotsman 2000], in order to evaluate properly their compression algorithm, introduce the *Geometric Laplacian* (GL), which measures the smoothness of a vertex v :

$$GL(v) = v - \frac{\sum_{i \in n(v)} l_i^{-1} v_i}{\sum_{i \in n(v)} l_i^{-1}} \quad (1)$$

where $n(v)$ is the set of indices of the neighbours of v , and l_i the Euclidean distance from v to v_i . $GL(v)$ represents the difference vector between v and its new position after a Laplacian smoothing step, thus it represents a kind of measure of roughness: the higher it is, the rougher is the surface around v . With the same idea, Drelie Gelasca et al. [Drelie Gelasca et al. 2005] propose a perceptual metric based on global *roughness* variation, to measure the quality of a watermarked mesh. They define the *roughness* as the variance of the geometric distances between a 3D model and its smoothed version, similarly to the *Geometric Laplacian* from Karni and Gotsman [Karni and Gotsman 2000]. Wu et al. [Wu et al. 2001] and Corsini et al. [Corsini et al. 2005] present a similar *roughness* measure based on the variance of the dihedral angles of edges surrounding a vertex.

The main problems of these existing roughness measures are that:

- They depend on the connectivity of the input mesh since they consider only the direct neighbours of each vertex to calculate their statistics. Particularly, they do not depend on a desired scale whereas the perception of a given object depends

on its distance from the camera, moreover the masking effect is quite linked with the frequency of the roughness. Thus a correct measure has to be driven by a scale parameter.

- Psychovisual researchers propose that there exist three principal relevant categories of regions in an image or a 3D object, they are presented on figure 1: Edge (like the ear), rough (like the ankle), and smooth (like the chest) regions. These categories are associated with different *masking* degrees, indeed a rough region exhibit a high degree of masking, whereas a geometric change on edge or smooth regions is much more visible. Existing measures often confuse rough regions and edge regions which is quite critical since their masking degrees are very different.

In this context we present a robust roughness measure for polygonal meshes which is illustrated on Figure 3. For each vertex, the corresponding roughness is processed basically by computing an asymmetric difference between local average curvatures computed on the original mesh and on a smoothed version. The curvature is computed and averaged over sizeable local windows of the mesh. Indeed our roughness measure is associated with a scale parameter which determines the size (i.e. the frequency) of the details that have to be considered as *noise* and that can lead to a masking effect for a given signal to hide. The main idea is that average curvatures computed over local windows aim at detecting regions associated with high geometric variations. However these variations can be caused by noise but also by the presence of salient features (edge regions). On the smoothed version of the object, the geometric noise disappears while salient features are preserved. Thus by computing curvature difference between original and smoothed versions, we accurately differentiate the real geometric noise (i.e. the roughness) from salient (edge) parts.

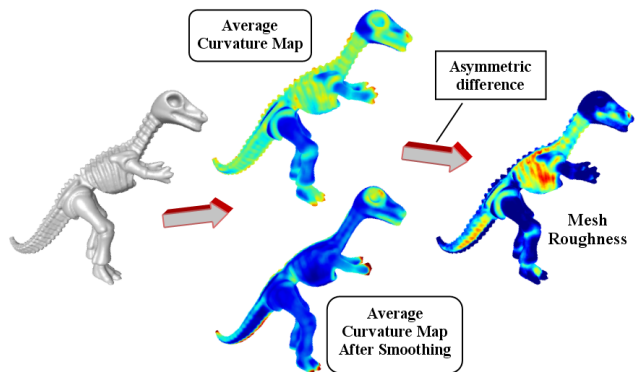


Figure 3: Overview of our roughness calculation algorithm.

2 Curvature calculation

In order to define a measure of roughness (or noise), the first step is to find a 3D measure analogous to the 2D concept of luminance of an image. The geometric information (i.e. the coordinates of the vertices) does not seem to be relevant for this objective since the human eye is not really sensitive to this information. Several authors have considered discrete curvature for 3D perceptual measures [Lee et al. 2005; Kim et al. 2002], indeed this information well describes the visual characteristics of a 3D model. In particular curvature variations strongly influence the intensity image coming from the rendering of the object.

A triangle mesh is a piecewise linear surface, thus the calculation of its curvature is not trivial. We have implemented the work of Cohen-Steiner et al. [Cohen-Steiner and Morvan 2003], based on the Normal Cycle, to estimate the mean curvature at each vertex of the mesh. This estimation procedure relies on solid theoretical

foundations and convergence properties. In order to remain independent of the connectivity of the input mesh, the tensor is averaged over a geodesic region, like in [Alliez et al. 2003].

3 Definition of a 3D local window

The proposed measure is based on a scale parameter which determines the frequencies that have to be considered as roughness noise. In order to establish this scale parameter we define the notion of *local window* of a mesh. However, even if the concept of *local window* is trivial in the field of 2D image (a squared neighbourhood for each pixel for instance), it becomes quite more complex for 3D objects with non-regular connectivity. For each vertex v , we define its associated *local window*, as the connected set of vertices belonging to the sphere with centre v and radius r (see blue points in figure 4.a). We also integrate intersections between this sphere and edges of the mesh, we call them *edge points* (see green points in figure 4.a), their curvature value is interpolated from their neighbours ones. In figure 4.b, the *edge point* v_e is added to the local window of vertex v , and its curvature $C(v_e)$ is calculated as follows:

$$C(v_e) = \frac{d_2}{d_1 + d_2} C(v_1) + \frac{d_1}{d_1 + d_2} C(v_2) \quad (2)$$

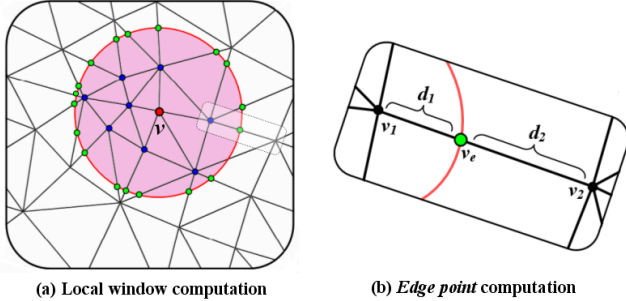


Figure 4: (a) Example of local window computation, for a vertex v . (b) Example of *edge point*.

4 Adaptive Smoothing

Our algorithm is based on a curvature difference between the original object and a smoothed version. This smoothing must be linked with a scale parameter, however much of existing algorithms do not provide such criterion; their smoothing effect is rather linked to the neighbourhood connectivity of the vertices and highly depends on the mesh density. Figure 5 illustrates this problem: the top row presents two versions of the same shape, respectively associated with 5K vertices (left) and 42K vertices (right) while the middle row presents results after 15 iterations of Laplacian smoothing. Resulting shapes are highly different, the left model is far more smoothed than the right one.

We have defined an Adaptive Smoothing that depends on a scale parameter while being independent of the object connectivity; it is derived from the two-step Taubin filter [Taubin 1995]. For this task we associate our smoothing filter to a ϵ scale factor. For a vertex v , its smoothed position v' is defined in two steps as follows:

$$v^m = (v + \lambda \cdot s(v)) \quad \text{with } s(v) = \frac{1}{n} \sum_{e_i \in x} (e_i - v)$$

$$v' = (v^m + \mu \cdot s(v^m))$$

with e_i the i^{th} edge point (in green in figure 4.a) of the local window x of v associated with the radius $r = \epsilon$. n is the number of edge points of this local window. λ and μ are respectively fixed to 0.6307 and -0.6732 (values originally suggested by Taubin). Figure 5, bottom row, illustrates the results of our adaptive smoothing (5

iterations) on the two objects associated with different mesh densities. Resulting shapes are basically identical, thus our smoothing is really independent of the mesh density. Moreover only a small number of iterations is necessary since our algorithm converges quickly toward a stable smoothed shape, which depends only on the ϵ scale factor.

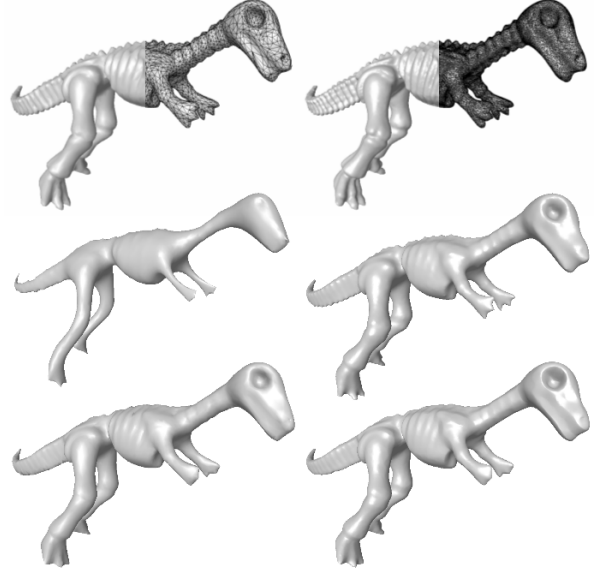


Figure 5: *Top row*: The Dyno shape associated with different mesh densities: 5K vertices (left) and 42K vertices (right). *Middle row*: Results after 15 iterations of Laplacian smoothing. *Bottom row*: Results after 5 iterations (convergence) of our Adaptive Smoothing.

5 The roughness measure

Our roughness measure is the following:

1. The 3D object is smoothed using our Adaptive Smoothing associated with an ϵ scale parameter (see section 4).
2. The curvature of each vertex of the two meshes (original and smoothed) is calculated (see section 2).
3. An average curvature value is processed for each vertex, it corresponds to the mean of the curvature of all vertices from its local window. The radius of local windows is set to 2ϵ . This radius size allows to correctly detect the noise that has been suppressed by the ϵ scale smoothing from (1).
4. We construct the Roughness Map by processing, for each vertex v_i , an asymmetric difference between its average curvature values on the original and the smoothed objects.

$$R(v_i) = AC(V_i) - AC(V_i^s) \quad \text{if } AC(V_i) > AC(V_i^s)$$

$$R(v_i) = 0 \quad \text{else}$$

This asymmetric difference permits to detect rough regions where the original average curvature $AC(v_i)$ (which represents a kind of *entropy*) is higher than the one $AC(v_i^s)$ of the smoothed version, while preventing the false detection of edge features (like the ear of Armadillo on figure 1).

6 Results and application to masking

Figures 1, 3 and 6 respectively illustrate the roughness maps of 3D objects: Armadillo (40K vertices), Dyno (42K vertices), Lion (39K vertices) and Bimba (9K vertices), with ϵ fixed to 1% of the length of the cubic bounding box of the models. In all of these examples,

noised regions are well detected, while smooth and edge parts are associated with very low roughness values.

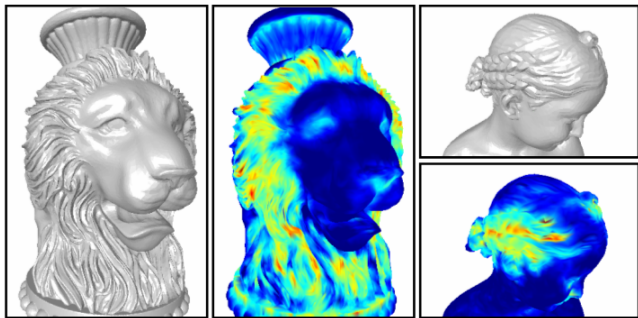


Figure 6: Roughness maps of the Lion (left) and Bimba (right) objects.

One interesting application of our roughness estimator is the masking effect. Indeed a rough region will be able to mask some geometric perturbations much better than a smooth surface (see figure 2). Hence we can imagine developing adaptive compression or watermarking algorithms where the compression artifacts or the watermark strength could be concentrated on rough parts where geometric modifications are nearly invisible.

In order to demonstrate this principle we have classified (K-means algorithm) the roughness of different objects into two clusters: Rather rough vertices and rather smooth vertices. We have then applied a random noise only on vertices from smooth and rough clusters respectively. Figure 7 illustrates this experiment for the Lion object. The noise strength is the same for both cases: 0.15% of the length of the cubic bounding box of the model. The object from the right, noised on rough parts, is associated with a higher Hausdorff geometric mean distance from the original than the object from the middle, noised on smooth parts (0.71×10^{-3} vs 0.66×10^{-3}). However the visual distortion is far less visible for the right object thanks to the masking effect. This masking is confirmed by the perceptual distortion measure *MSDM* [Lavoué et al. 2006] which provides an approximation of the subjective visual similarity. This value tends toward 1 (theoretical limit) when the measured objects are visually different and is equal to 0 for identical ones. This perceptual metric gives a smaller value (and thus a higher perceptual similarity) for the object noised on rough parts (0.36 vs 0.42).

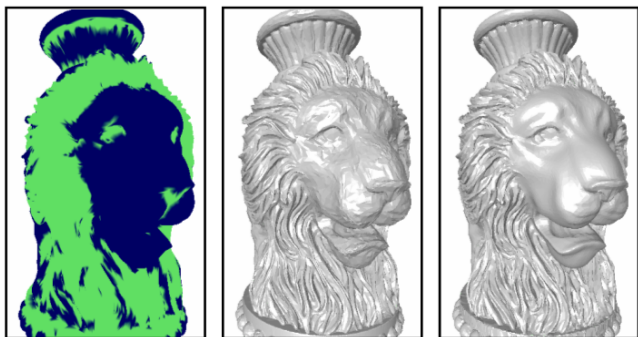


Figure 7: Left: Roughness classification into two clusters: rough vertices are in green and smooth ones are in blue. Middle: Random noise on smooth regions. Right: Random noise on rough regions.

7 Conclusion

We have presented a robust roughness measure for 3D meshes which does not depend on the connectivity but is driven by a scale parameter defining the size of the noise that we aim to detect. This estimator well detects rough regions while not confusing them with

edge regions or salient parts. We have presented an application to Masking: A rough region is able to hide small geometric distortions much better than a smooth one. This measure could be advantageously integrated to compression or watermarking algorithms in order to concentrate artefacts on rough parts of the object which exhibit a high masking degree.

References

- ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics* 22, 3, 485–493.
- COHEN-STEINER, D., AND MORVAN, J. 2003. Restricted delaunay triangulations and normal cycle. In *19th Annu. ACM Symp. Comput. Geom.*
- CORSINI, M., DRELIE GELASCA, E., AND EBRAHIMI, T. 2005. A multi-scale roughness metric for 3d watermarking quality assessment. In *Workshop on image analysis for multimedia interactive services.*
- DALY, S. 1993. *The visible differences predictor: An algorithm for the assessment of image fidelity.* MIT Press, Cambridge, MA, 179–206.
- DRELIE GELASCA, E., CORSINI, M., AND EBRAHIMI, T. 2005. Objective evaluation of the perceptual quality of 3d watermarking. In *IEEE International Conference on Image Processing*, 241–244.
- FERWERDA, J., PATTANAIK, S., SHIRLEY, P., AND GREENBERG, D. 1997. A model of visual masking for computer graphics. In *ACM Siggraph*, 143–152.
- HOWLETT, S., HAMILL, J., AND O’SULLIVAN, C. 2005. Predicting and evaluating saliency for simplified polygonal models. *ACM Transactions on Applied Perception* 2, 3, 286–308.
- KARNI, Z., AND GOTSMAN, C. 2000. Spectral compression of mesh geometry. In *ACM Siggraph*, 279–286.
- KIM, S., KIM, S., AND KIM, C. 2002. Discrete differential error metric for surface simplification. In *Pacific Graphics*, 276–283.
- LAVOUÉ, G., DRELIE GELASCA, E., DUPONT, F., BASKURT, A., AND EBRAHIMI, T. 2006. Perceptually driven 3d distance metrics with application to watermarking. In *SPIE Applications of Digital Image Processing XXIX*, vol. 6312.
- LEE, C., VARSHNEY, A., AND JACOBS, D. 2005. Mesh saliency. In *ACM Siggraph*, 659–666.
- LUEBKE, D., AND HALLEN, B. 2001. Perceptually driven simplification for interactive rendering. In *Eurographics Workshop on Rendering Techniques*, 223–234.
- REDDY, M. 2001. Perceptually optimized 3d graphics. *IEEE Computer Graphics and Applications* 21, 5, 68–75.
- RONDAO-ALFACE, P., DE CRAENE, M., AND MACQ, B. 2005. Three-dimensional image quality measurement for the benchmarking of 3d watermarking schemes. In *SPIE Security, Steganography, and Watermarking of Multimedia Contents*, 230–240.
- TAUBIN, G. 1995. A signal processing approach to fair surface design. In *ACM Siggraph*, 351–358.
- WU, J.-H., HU, S.-M., SUN, J.-G., AND TAI, C.-L. 2001. An effective feature-preserving mesh simplification scheme based on face constriction. In *Pacific Graphics*, 12–21.