

# A Local Roughness Measure for 3D Meshes and its Application to Visual Masking

GUILLAUME LAVOUÉ  
Université de Lyon, CNRS  
INSA-Lyon, LIRIS

3D models are subject to a wide variety of processing operations such as compression, simplification or watermarking, which may introduce some geometric artifacts 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 one visual pattern can hide the visibility of another. In this context we introduce an algorithm for estimating the *roughness* of 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. The accuracy and the robustness of our measure, together with its relevance regarding visual masking have been demonstrated through extensive comparisons with state-of-the-art and subjective experiment. Two applications are also presented, in which the roughness is used to lead (and improve) respectively compression and watermarking algorithms.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.3.m [**Miscellaneous**]: Perception

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: 3D mesh, Roughness, Curvature, Masking, subjective evaluation

## ACM Reference Format:

Lavoué, G. 2009. A local roughness measure for 3D meshes and its application to visual masking. *ACM Trans. Appl. Percept.* 5, 4, Article 21 (January 2009), 23 pages. DOI = 10.1145/1462048.1462052 <http://doi.acm.org/10.1145/1462048.1462052>

## 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, or watermarking. These operations often introduce visual artifacts which deteriorate the visual quality of the shape. Figure 1 illustrates some examples of these artifacts: watermarking (method from Wang et al. [2007]) introduces both low frequency and high frequency distortions, simplification induces a kind of smoothing effect, while

Authors' address: G. Lavoué, LIRIS UMR 5205 CNRS, INSA-Lyon, F-69621, France.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2009 ACM 1544-3558/2009/01-ART21 \$5.00 DOI 10.1145/1462048.1462052 <http://doi.acm.org/10.1145/1462048.1462052>

ACM Transactions on Applied Perception, Vol. 5, No. 4, Article 21, Publication date: January 2009.

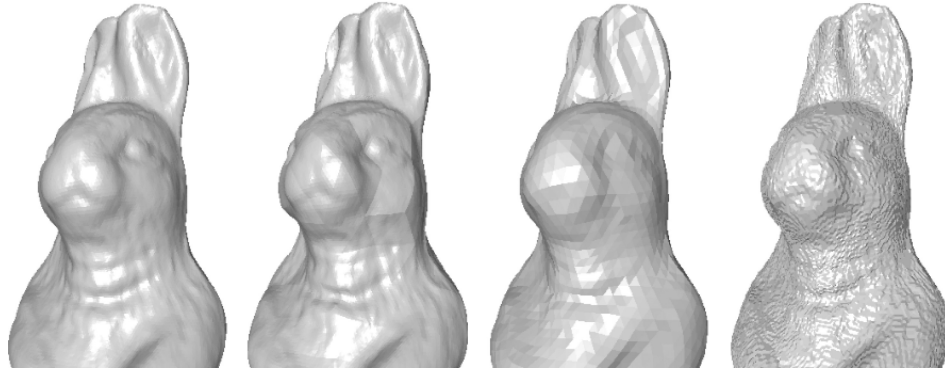


Fig. 1. Some examples of artifacts introduced by common processing operations for the head of the Bunny mesh. *From left to right*: Original model (71K vertices), result after watermarking (method from Wang et al. [2007]), result after simplification (from 71K vertices to 4.5K vertices) and result from quantization (9 bits precision).

quantization (i.e. compression) leads to a certain form of noise, but its effect is somewhat complicated. The objective of these algorithms is obviously to minimize the visual degradations, but the main problem is that they are mainly driven by geometric criteria, which do not correlate with the human visual perception.

Since these processes are generally committed for human centred applications, it seems relevant to incorporate principles of the human visual system in their realization.

Our idea is based on a weakness of the human visual system, which cannot distinguish a small distortion if it is located on a rough (i.e. noised) area. Hence we present a simple roughness measure, allowing to locate the regions where geometric artifacts would be near invisible. This measure is able to drive compression, simplification, or watermarking algorithms by concentrating the distortion in such rough areas.

The next section describes the related work about perceptually adaptive graphics and perceptual models in computer graphics. Section 3 provides an overview and details about the different steps of our roughness measure. Section 4 presents some visual results, comparisons with state-of-the-art robustness experiments, and a subjective evaluation of the visual masking. Finally Section 5 illustrates two applications: compression and watermarking, in which our roughness estimator is used to improve the results.

## 2. PREVIOUS WORK

Many computational models of the human visual system have been developed in the field of 2D image processing particularly to evaluate the perceptual similarity between two images; a review has been done by Eckert and Bradley [1998]. Basically there exist two different approaches: *computational* and *ad-hoc*.

*Computational* metrics, like the Visible Difference Predictor (VDP) [Daly 1993], consist in complex numerical models which rely on some psychophysical and physiological evidences like the *masking effect*; this mechanism defines the fact that a signal can be masked by the presence of another signal with similar frequency or orientation.

*Ad-hoc* metrics consider simpler mathematical measures, intuitively relating to the visual perception, and/or introduce penalties for specific artifacts.

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. In fact these methods are based on 2D image perceptual models described in the previous paragraph. Reddy [2001], Luebke and Hallen [2001] and Williams et al. [2003] present such view-dependent simplification algorithms. In a different way, Bolin and Meyer [1998] use perceptual models to optimize the sampling for ray tracing algorithms. Most of these works concern off-line rendering; Dumont et al. [2003] present a view dependent simplification algorithm for real time rendering based on the worst cases imperceptible contrast and spatial frequency changes. Although the existing work on realistic rendering is mostly based on 2D image metrics, several authors have considered kinds of 3D metrics like Tian and AlRegib [2004] and Pan et al. [2005]. Their metrics rely respectively on geometry and texture deviations [Tian and AlRegib 2004] and on texture and mesh resolutions [Pan et al. 2005].

Little attention has been paid to the use of perceptual models for standard 3D mesh processing. Some *ad-hoc* metrics have been proposed to guide mesh simplification [Kim et al. 2002], to evaluate compression [Karni and Gotsman 2000], or watermarking algorithms [Rondao-Alface et al. 2005; Corsini et al. 2007], or to 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. [2005] which aims 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. [2005] determine the saliency through an eye tracking system.

Our objective is somehow dual to these works on saliency: 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. This perceptual phenomenon has been studied for many years by physiologists and psychologists [Campbell and Kulikowski 1966; Harmon and Julesz 1973]. Qu and Meyer [2006] have proposed a remeshing algorithm driven by the masking effect produced by a 2D texture map associated to the mesh. Differently, we consider the masking effect produced by the *geometric* noise. 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. [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 perception. A lot of roughness-like measures have been considered in most of existing perceptual 3D metrics. 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

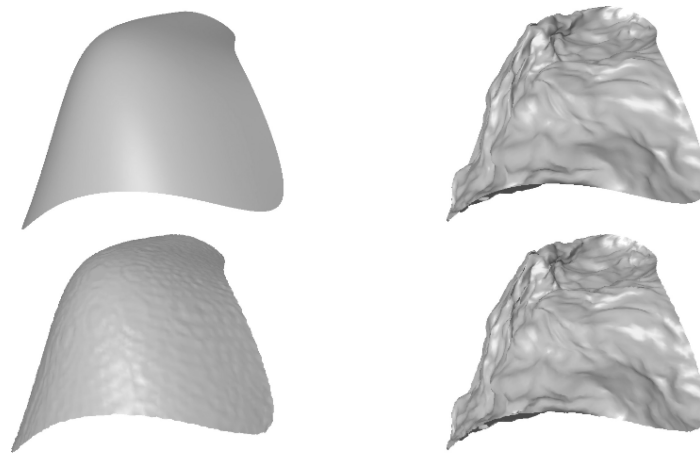


Fig. 2. Effect of roughness for 3D masking. The upper images are original while lower ones are quantized on 8 bits. The quantization artifacts are much more visible on the smooth surface than on the rough one because of the masking effect.

the same idea, Corsini et al. [2007] 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 [2000]. Wu et al. [2001], and once again Corsini et al. [2007], present a similar *roughness* measure based on the variance of the dihedral angles of edges surrounding a vertex. Lastly, Lavoué et al. [2006] define the roughness as a measure of curvature variance.

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 scale parameter, whereas the perception of a given object depends on such scale information like its distance from the camera, for instance; 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 [Eckert and Bradley 1998] which can be generalized to 3D objects (see figure 3): edge (like the borders of the ears), rough (like the ankles), and smooth (like the chest) regions. These categories are associated with different *masking* degrees; indeed a rough region exhibits 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, this misclassification is quite critical since their masking degrees are very different.

### 3. OUR ROUGHNESS MEASURE

#### 3.1 Overview

We present a robust roughness measure for 3D polygonal meshes which is illustrated on Figure 4. 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

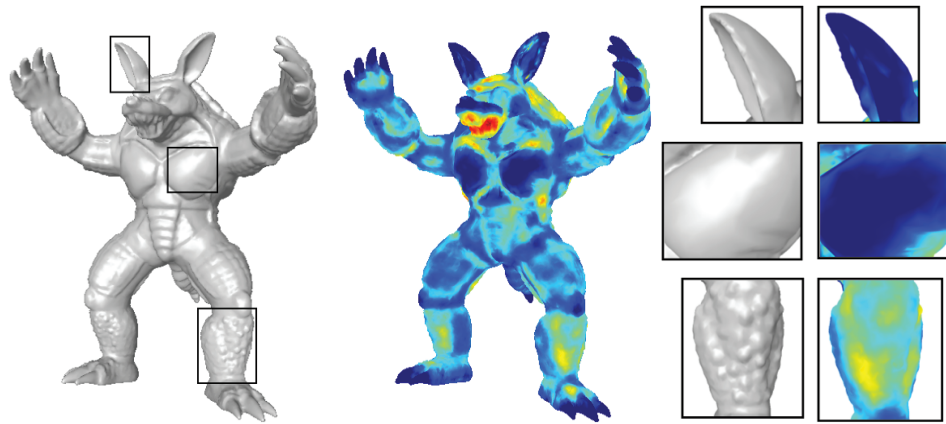


Fig. 3. The Armadillo 3D model and its *roughness* map. Warmer colors (reds, yellows, greens) illustrate high roughness values while cooler colors (dark blue) illustrate rather smooth regions. The different categories of regions are illustrated on the right: Edge (top), smooth (middle) and rough (bottom) regions.

hide. The basic ideas behind this mechanism are the following:

- (1) The average curvature values, computed over local windows, aim at detecting regions associated with high geometric variations.
- (2) Basically these variations can be produced by three different kinds of regions: really rough parts (like the ribs of the Dyno), edge features (like the borders of the eyes for instance), or just highly curved smooth regions (like the arms).
- (3) On the smoothed version of the object, the geometric noise (i.e roughness) disappears while edge features and highly curved parts are well preserved. Thus by computing the curvature difference between original and smoothed versions, we accurately differentiate the real geometric roughness from edge parts and smooth regions.

One could object that, in practice, the roughness which is *perceived* by a human observer also depends on the viewpoint and the lighting conditions. Indeed, a view-dependent roughness measure may be more efficient and more precise than a purely geometric measure like ours. However, most of existing compression, watermarking, or simplification algorithms are applied in a purely geometric way without viewpoint considerations (except view-dependent simplification dedicated to real-time rendering), hence our measure has to follow this purely geometric framework to be able to efficiently drive these processes. Moreover, we believe that the perceptual features and behavior of a 3D model have to be extracted from the 3D geometry itself and cannot be correctly captured through image representations; besides Rogowitz and Rushmeier [2001] have demonstrated that image quality metrics cannot be used reliably for assessing 3D object quality.

We also have to precise here, that our measure only takes into account the geometric information of the object. We do not consider some possibly associated properties (e.g. a texture map) that could also have influence on the visual appearance of the object. Some works have already been done about such nongeometric factors, for instance about *texture* masking [Ferwerda et al. 1997; Qu and Meyer 2006].

### 3.2 Curvature Calculation

In order to define a measure of roughness (i.e. 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

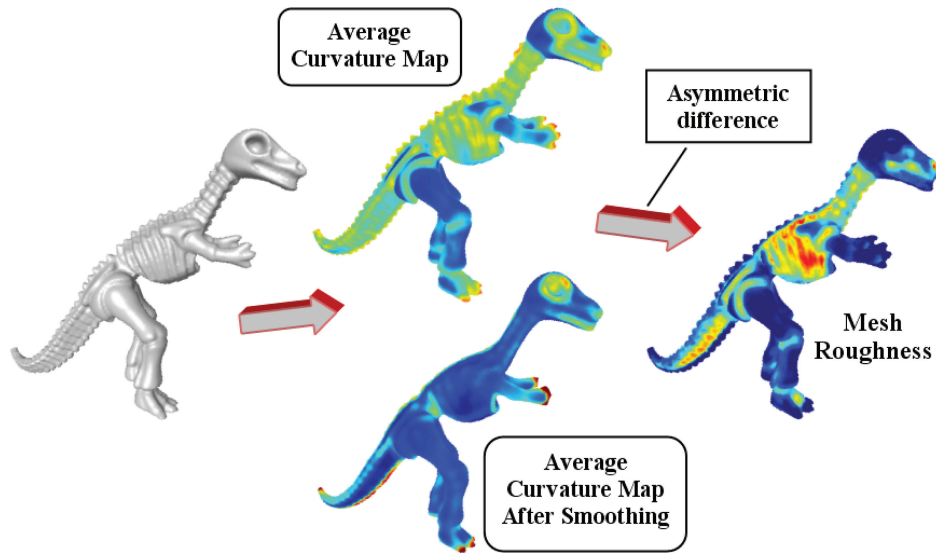


Fig. 4. Overview of our roughness calculation algorithm.

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, variations of the maximum curvature 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 and Morvan [2003], based on the Normal Cycle, to estimate the curvature tensor 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 integrated over a geodesic region, like in [Alliez et al. 2003]. For each vertex, the curvature tensor is calculated and then the principal curvature values  $k_{min}$ ,  $k_{max}$  are extracted, they correspond to the eigenvalues of the curvature tensor. For our roughness estimation algorithm, we consider the maximum curvature  $k_{max}$ , since this value really reflects the bumpiness of the surface. An example of maximum curvature map is given on figure 6, on the left.

### 3.3 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 nonregular connectivity. For each vertex  $v$ , we define its associated *local window*  $W_r(v)$ , as the connected set of vertices belonging to the sphere with centre  $v$  and radius  $r$  (see blue points in figure 5(a)). We also integrate intersections between this sphere and edges of the mesh, we call them *edge points* (see green points in figure 5(a)), their curvature value is interpolated from their neighbours' ones. For instance in figure 5(b), the *edge point*  $v_e$  is added to the local window of vertex  $v$ , and its curvature value  $k_{max}(v_e)$

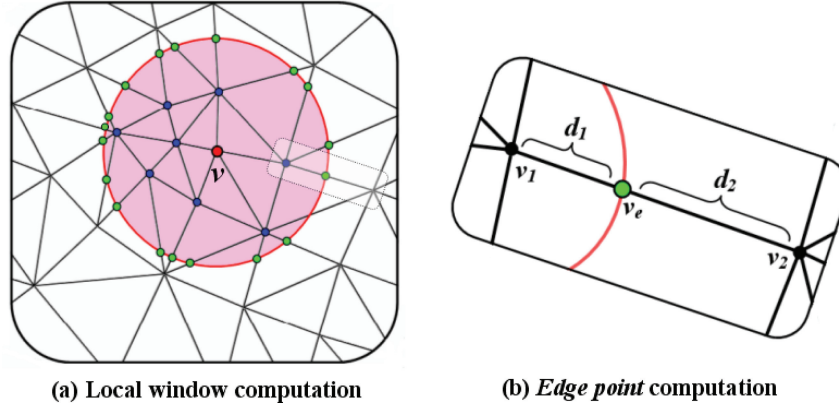


Fig. 5. (a) Example of local window computation, for a vertex  $v$ . (b) Example of *edge point*.

is calculated as follows:

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

Including *edge points* allows us to overcome the intrinsic irregularity of the mesh. We can then calculate the average curvature value  $kav$  of each vertex: it corresponds to the average of the curvature values  $kmax$  of the vertices belonging to its local window:

$$kav(v) = \frac{1}{|W_r(v)|} \sum_{v_i \in W_r(v)} kmax(v_i). \quad (3)$$

Figure 6 illustrates the  $kmax$  and  $kav$  maps for the Dyno model, corresponding to an average over local windows of size  $r = 2\%$  of the side-length of the cubic bounding box of the model.

### 3.4 Adaptive Smoothing

Our algorithm is based on the average curvature difference between the original object and a smoothed version. This smoothing step has to be linked with a scale parameter, however, much of existing algorithms do not provide such criterion: usually, for a given vertex, the smoothing mask rather relies on the connectivity of its neighborhood; hence, the smoothing effect highly depends on the mesh sampling density. Figure 7 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 define an *adaptive smoothing* that depends on a scale parameter while being independent of the object connectivity; the key idea is to construct the smoothing masks regarding local windows (defined in the previous section), instead of the direct neighborhoods of the vertices. Our smoothing algorithm is derived from the two-step Taubin filter [Taubin 1995] and is associated with a  $\epsilon$  scale factor. For a vertex  $v$ , its smoothed position  $v'$  is defined in two steps as follows:

$$v^m = (v + \lambda \cdot s(v)) \quad \text{with } s(v) = \frac{1}{n} \sum_{e_i \in W_r(v)} (e_i - v) \quad (4)$$

$$v' = (v^m + \mu \cdot s(v^m)), \quad (5)$$

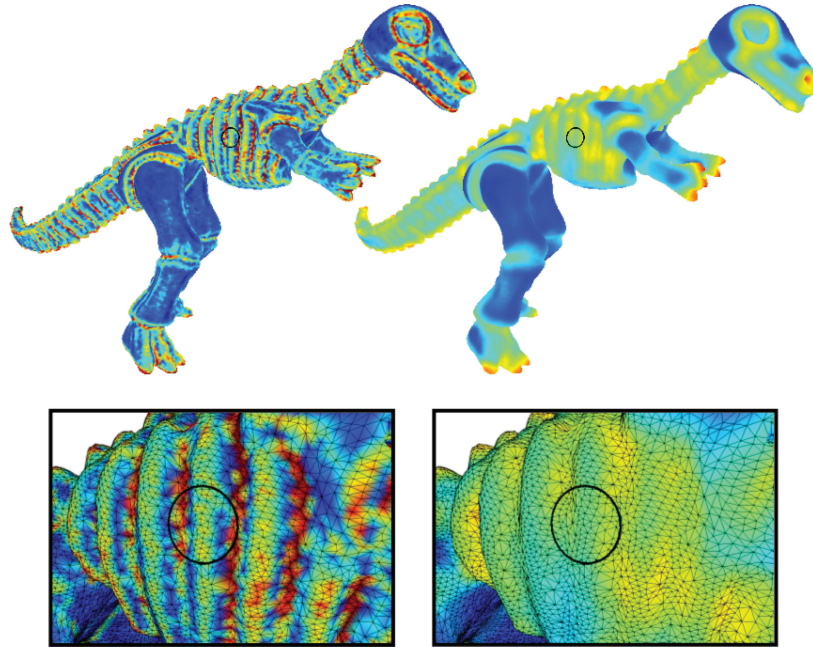


Fig. 6. *Left*: The maximum curvature ( $k_{max}$ ) map of the Dyno model, and a zoom on the torso part. *Right*: The average curvature ( $k_{av}$ ) map; for each vertex the  $k_{max}$  values were averaged over a local window of size  $r = 2\%$  of the length of the cubic bounding box. An example of local window is illustrated by the black circle.

with  $e_i$  the  $i^{th}$  edge point (in green in figure 5(a)) of the local window  $W_r$  of  $v$  associated with the radius  $r = \epsilon$ .  $n$  is the number of edge points of this local window.  $\lambda$  and  $\mu$  are respectively fixed to 0.6307 and  $-0.6732$  (values originally suggested by Taubin). Figure 7, bottom row, illustrates the results of our adaptive smoothing (five 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 are necessary since our algorithm converges quickly toward a stable smoothed shape, which depends only on the  $\epsilon$  scale factor.

It is important to note that this smoothing algorithm induces a little *thinning* effect (like the original Taubin filter), therefore, the salient edge features (i.e. the strong ridges, like the ears of Armadillo or the hands and feet of Dyno) are thinned and thus become sharper. This sharpening effect allows us to differentiate these regions from rough ones since their curvature increases after being smoothed. However this phenomenon appears only on very strong edge features and not on every sharp parts of the mesh. One solution to improve this sharp edge detection would be to consider some feature preserving smoothing like the work from Jones et al. [2003].

### 3.5 The Roughness Measure

Our roughness measure is computed as follows:

- (1) The 3D object is smoothed using our *adaptive smoothing* filter associated with a  $\epsilon$  scale parameter (see Section 3.4).
- (2) The curvature  $k_{max}$  of each vertex from the two meshes (original and smoothed) is calculated (see Section 3.2).



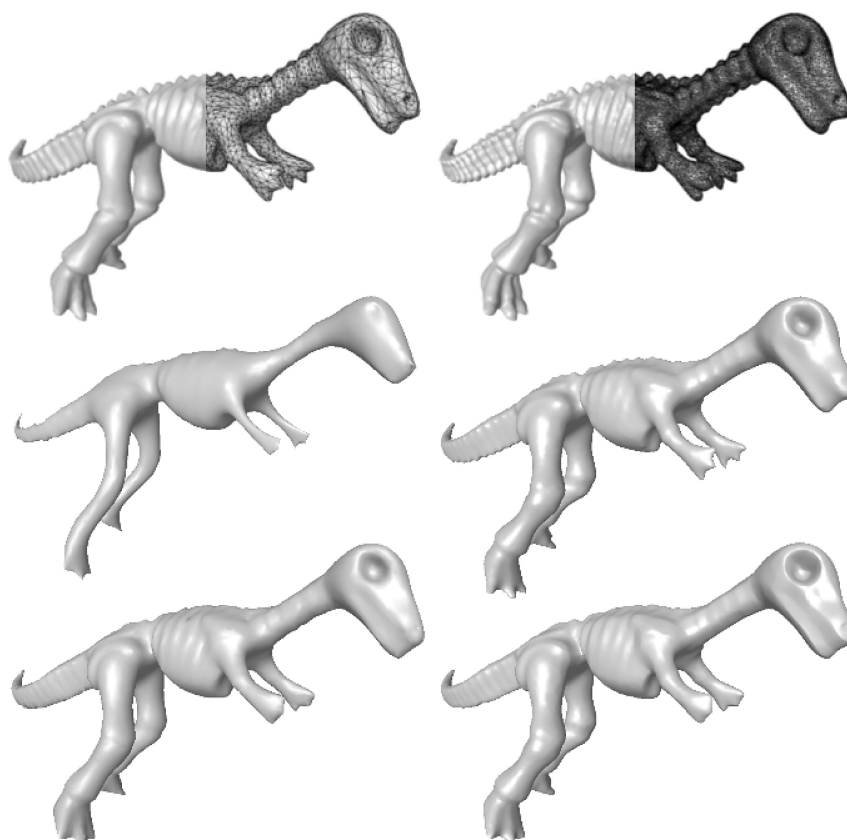


Fig. 7. *Top row*: The Dyno shape associated with different mesh densities: 5K vertices (left) and 42K vertices (right). *Middle row*: Results after 15 iterations of Laplacian smoothing. *Bottom row*: Results after 5 iterations of our *adaptive smoothing*.

- (3) The average curvature value  $kav$  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  $2\epsilon$ . This radius size allows us to correctly detect the noise that has been suppressed by the  $\epsilon$  scale smoothing from (1).
- (4) The roughness values are computed by calculating, for each vertex  $v$ , an asymmetric difference between its average curvature values on the original and the smoothed versions

$$R(v) = kav(v) - kav(v^s) \quad \text{if } kav(v) > kav(v^s) \quad (6)$$

$$R(v) = 0 \quad \text{else.} \quad (7)$$

This asymmetric difference permits us to detect rough vertices where the original average curvature  $kav(v)$  (which represents a kind of *entropy*) is higher than the one  $kav(v^s)$  on the smoothed version. Moreover, it prevents the false detection of edge features (like the borders of the ears of Armadillo on figure 3 for instance); indeed, after smoothing certain sharp edge features may exhibit a higher  $kav$  value than on the original version (see end of Section 3.4), in that case, we fix the roughness value to zero (see equation 7).

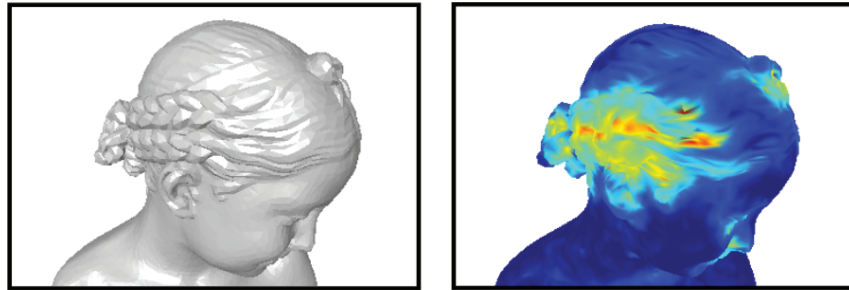


Fig. 8. Roughness map of the Bimba model for  $\epsilon = 1\%$  of the length of the cubic bounding box.

## 4. EXPERIMENTS AND RESULTS

Figures 3, 4, 8, and 9, respectively, illustrate the roughness maps of the following 3D objects: Armadillo (40K vertices), Dyno (42K vertices), Lion Head (39K vertices) and Bimba (9K vertices). Warmer colors (reds, yellows, greens) illustrate high roughness values while cooler colors (dark blue) illustrate rather smooth regions. In all of these examples, rough regions are well detected, while smooth and edge parts are associated with very low roughness values. The processing time depends on the number of vertices and on the value of the  $\epsilon$  scale parameter. For our examples, it varies from 2.5s (Bimba model, with  $\epsilon = 1\%$ ) to 110s (Lion Head, with  $\epsilon = 3\%$ ). The main time consuming step is the calculation of the local windows, particularly for high  $\epsilon$  values.

### 4.1 The Effect of the $\epsilon$ Scale Parameter

Figure 9 shows the roughness maps of the Lion Head model for different values of  $\epsilon$ . The complex geometric texture of the *mane* of the Lion can be perceived differently by a human observer, depending on the distance of the object to the camera, for instance. This characteristic is particularly well reflected by our roughness estimator. For a small  $\epsilon$  value (middle figure), some features of the *mane* (at the bottom left of the head) appear in blue because they are not considered as rough parts, but rather as combinations of smooth and edge regions. On the contrary, for a large  $\epsilon$  value (right figure) these regions appear as rough (warmer colours). Basically a small  $\epsilon$  leads to a roughness map that corresponds to the roughness perceived by an observer who is very close to the object. On the contrary a large  $\epsilon$  corresponds to an object which will be quite distant from the observer; the theoretical upper limit is that the whole model is considered as rough, since at a certain distance, the object becomes too small to distinguish any structured information (there is only noise). In practice the range of available  $\epsilon$  for a given object is limited by the resolution of the model (it has no sense to consider an  $\epsilon$  much smaller than the size of the mesh facets) and by its complexity (a too large  $\epsilon$  leads to high calculation time).

### 4.2 Comparison with Existing Roughness Estimators

We have compared our roughness estimator with very recent algorithms from the state of the art: the two estimators M1 and M2 from [Corsini et al. 2007], respectively, based on dihedral angle variance and geometric distance variance from a smoothed version. We also compare with our previous estimator [Lavoué et al. 2006], based on curvature variance. Results are presented on Figure 10 which illustrates for each algorithm:

- a colorimetric representation, of which range depends on the method,
- the distribution of the roughness values from the minimum value to the maximum.

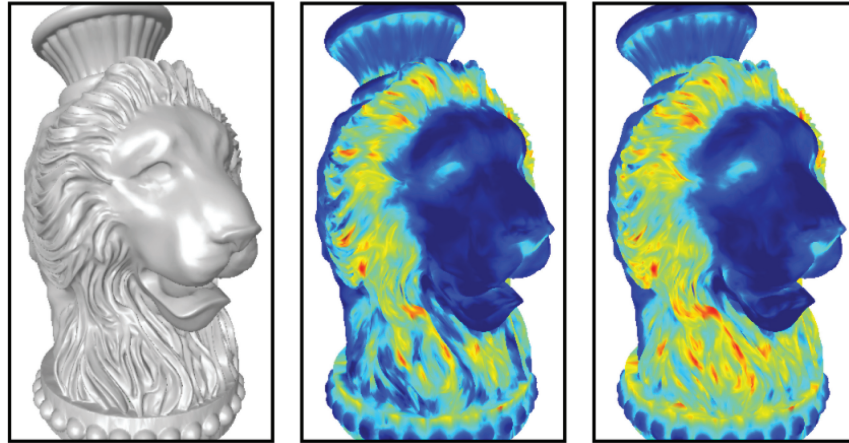


Fig. 9. Roughness maps of the Lion model for  $\epsilon = 1\%$  (middle) and  $\epsilon = 3\%$  of the length of the cubic bounding box.

The three methods from the state of the art all share the same problem: *edge* parts are misclassified as *rough* parts. For instance, the extremities of the hands and of the feet of the Dyno exhibit high roughness values for every method (except ours), whereas they are *edge* parts. That is the same problem for the vertebral column of the Dyno; indeed even if this region seems *bumpy* this is not a rough region, but rather a salient *edge* feature in the sense that it does not exhibit a high masking degree (i.e. even a small geometric distortion will be clearly visible). Our algorithm well differentiate these edge regions from rough ones.

The second problem with existing algorithms (particularly for those from Corsini et al. [2007]) is that the roughness distribution is very irregular. The histograms representing the distribution from the minimum value to the maximum value are not properly spread along the spectrum but are concentrated on the left side. This means that the estimator is not so robust and may produce very high values that can be inadequate for a further classification. On the contrary, our roughness measure is well spread along the spectrum of values. There is a high peak for the zero value because of our asymmetric difference formulation (see equation 7).

### 4.3 Robustness to Simplification

In order to check the independence of our roughness estimator regarding connectivity changes, we have tested its robustness to simplification. Figure 11 illustrates the maximum value and the distribution of the roughness for three algorithms, including ours, applied on the Dyno shape. These roughness statistics are processed for four versions of the shape associated with different sampling densities: From top to bottom, the numbers of vertices are respectively 30K, 20K, 10K, and 5K vertices. The global profiles of roughness distributions coming from methods M1 and M2 [Corsini et al. 2007] change in a significant manner depending on the sampling densities. Moreover, the maximum values fluctuate a lot: from the 30K version to the 5K version, they have increased respectively by 200% and 50% for M1 and M2. Our algorithm is much more robust, indeed the max value remains around 90 (the variation between high and low resolution versions is 17%). Moreover, the overall shape of the distribution remains approximatively the same.

This superiority of our measure is logical since it has been designed to be independent of the connectivity. On the contrary both methods from Corsini et al. [2007] are based on vertex neighborhoods and assume uniform-sampled models; however, the M2 method, which is multi-resolution, demonstrates a certain robustness.

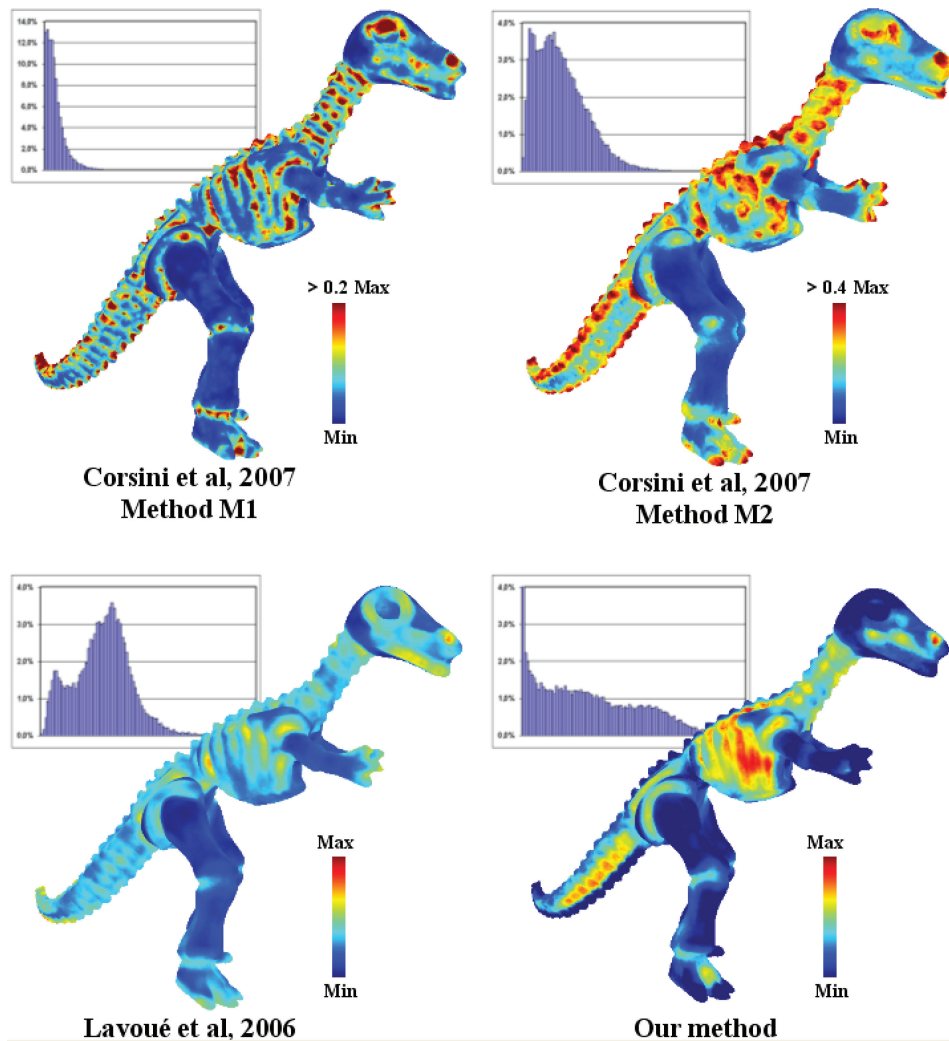


Fig. 10. Comparison of several roughness algorithms.

#### 4.4 Masking Effect Analysis

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 one (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 constructed an experimental corpus: we have added noise either on smooth or rough parts of some 3D models, then we have conducted two experiments to demonstrate the masking effect: (1) an objective evaluation of the corresponding visual degradations using an existing perceptual metric and (2) a subjective evaluation with human observers who have been asked to rate the visual degradations.

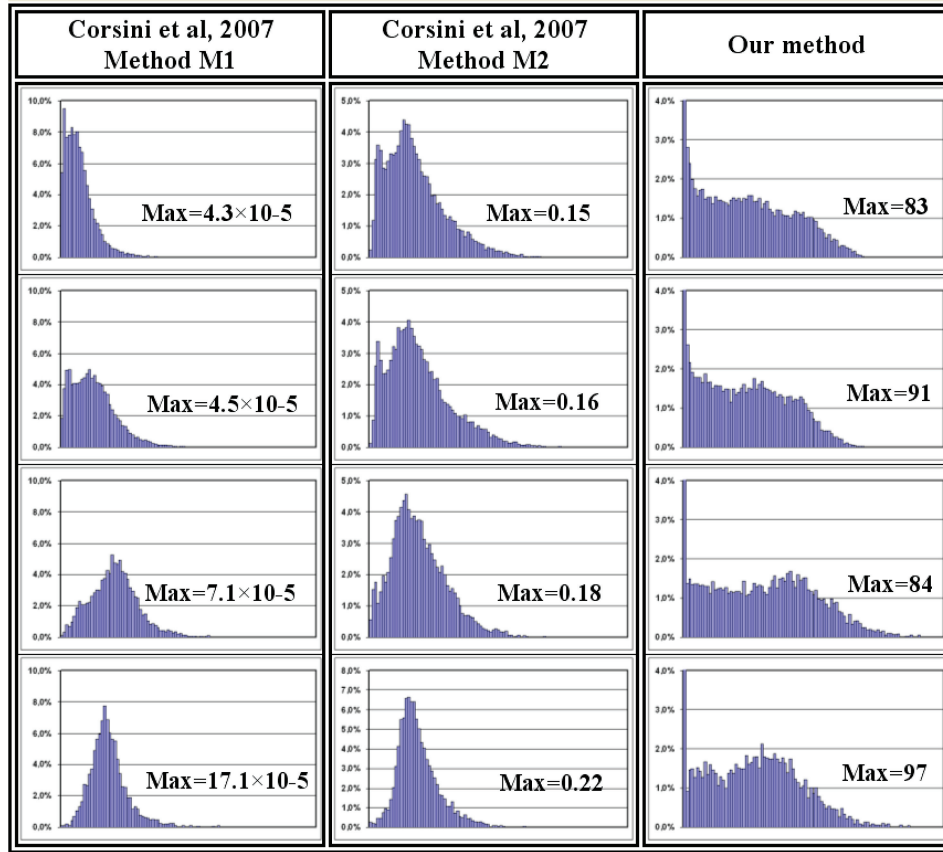


Fig. 11. Robustness of several roughness estimators against mesh simplification, for the Dyno shape. From top to bottom, the numbers of vertices are respectively 30K, 20K, 10K, and 5K.

4.4.1 *Database Construction.* The experimental corpus (i.e. the set of distorted 3D objects to evaluate) was constructed as follows: we have considered the models Armadillo, Dyno, Lion Head, and Bimba and then we have classified (K-means algorithm) their roughness into two clusters: rather rough vertices and rather smooth vertices (see Figure 12, on the left). We have then applied a random noise only on vertices from smooth and rough clusters respectively. The objective is to demonstrate that the perceived perturbation is less visible when the noise is applied on rough regions. In order to obtain a suitable evaluation of the masking effect, the noise was applied on rough and smooth regions with different strengths such as to obtain the same Root Mean Square error (RMS) in each case (this error reflects the amount of total *energy* of the added noise). In other words, if the total area of the rough region is larger than the area of the smooth one, the noise applied on the rough region will be lower (and vice versa). These noise distortions were applied according to three strengths (visually chosen): high, medium, and low. Hence, the experimental corpus contains 28 models (4 originals + 4 × 3 versions noised on smooth parts + 4 × 3 versions noised on rough parts). Figure 12 illustrates an example of this corpus. The model in the middle was noised on smooth parts while the model on the right was noised on rough parts (medium distortion). The noise strength is a little higher for smooth regions: (0.155% against 0.15% of the length of the cubic bounding box of the model). Both noised versions are associated

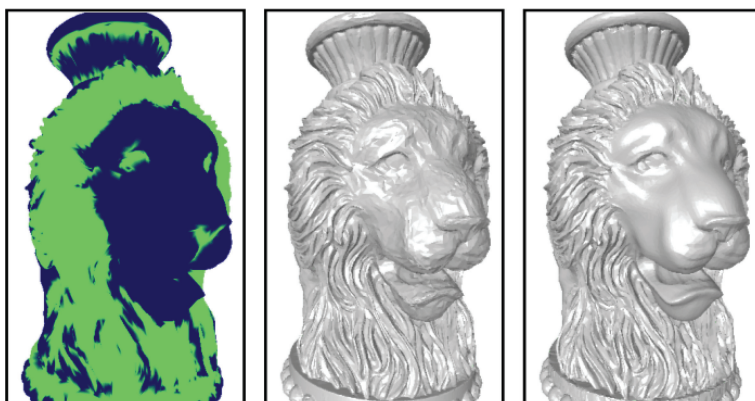


Fig. 12. *Left*: Roughness classification into two clusters: rough vertices are in green and smooth ones are in blue. *Middle*: Random noise on smooth regions. *Right*: Random noise on rough regions.

with the same RMS distance from the original model ( $1.04 \times 10^{-3}$ ). As expected, the visual distortion is far less visible for the object on the right thanks to the masking effect.

**4.4.2 Evaluation Using an Objective Perceptual Metric.** An easy way to capture the human perception without conducting a subjective experiment is to use an objective perceptual metric, like MSDM [Lavoué et al. 2006]. This metric was designed to reflect the perceptual distance between two 3D objects; this measure follows the concept of structural similarity recently introduced for 2D image quality assessment by Wang et al. [2004]. The local LMSDM distance between two mesh local windows  $x$  and  $y$  is defined as follows:

$$LMSDM(x, y) = (0.4 \times L(x, y)^3 + 0.4 \times C(x, y)^3 + 0.2 \times S(x, y)^3)^{\frac{1}{3}}. \quad (8)$$

L, C and S represent respectively curvature, contrast and structure comparison functions:

$$L(x, y) = \frac{\|\mu_x - \mu_y\|}{\max(\mu_x, \mu_y)}, \quad C(x, y) = \frac{\|\sigma_x - \sigma_y\|}{\max(\sigma_x, \sigma_y)}, \quad S(x, y) = \frac{\|\sigma_x \sigma_y - \sigma_{xy}\|}{\sigma_x \sigma_y}, \quad (9)$$

with  $\mu_x$ ,  $\sigma_x$  and  $\sigma_{xy}$ , respectively, the mean, standard deviation and covariance of the curvature over the mesh local windows. The global MSDM measure between two meshes  $X$  and  $Y$ , is defined by a Minkowski sum of their  $n$  local window distances:

$$MSDM(X, Y) = \left( \frac{1}{n} \sum_{j=1}^n LMSDM(x_j, y_j)^3 \right)^{\frac{1}{3}} \in [0, 1[. \quad (10)$$

Its value tends toward one (theoretical limit) when the measured objects are visually very different and is equal to zero for identical ones. Table I presents the MSDM results for some objects of the corpus (those associated with a medium strength). For each object, the perceptual measure gives a better similarity score (i.e. a lower value) when the noise is applied on rough regions. That somehow attests that our estimator well extracts rough regions associated with a high masking degree.

**4.4.3 Subjective Evaluation.** In order to fully demonstrate that our estimator successfully extracts high masking regions, we have also conducted a subjective experiment using human observers.

Watson et al. [2001] provide details about the different existing experimental measures for subjective evaluation: *naming times*, *rating* and *forced choice preference*. With rating, the observers assign to each

Table I. MDSM Evaluation of the Masking Effect of Rough Regions Estimated Using Our Algorithm

	Noise location	MSDM	RMSE ( $\times 10^{-3}$ )	Noise strength
Bimba (9K vertices)	Rough parts	0.263	1.21	0.24%
	Smooth parts	0.436	1.22	0.15%
Lion Head (39K vertices)	Rough parts	0.354	1.04	0.15%
	Smooth parts	0.395	1.04	0.155%
Dyno (20K vertices)	Rough parts	0.176	0.73	0.11%
	Smooth parts	0.260	0.72	0.10%
Armadillo (9K vertices)	Rough parts	0.217	0.66	0.10%
	Smooth parts	0.239	0.66	0.09%

Table II. Analysis of Variance (ANOVA) for Our Subjective Experiment

Source	SS	df	MS	F	p
Subjects	20.29	10	2.03	1.09	0.37
Error	468.95	253	1.85		
Total	489.24	263			

stimulus object a number reflecting its visual quality or visual distance with a reference object. Since it is suggested in CCIR Recommendation 500-3 [CCIR 1986], and like much of existing experimental assessment protocols [Pan et al. 2005; Lavoué et al. 2006; Corsini et al. 2007; Howlett et al. 2005] we have chosen this experimental measure.

The evaluation protocol is as follows: for each of the four models from the corpus (Armadillo, Dyno, Lion Head and Bimba), the corresponding six degraded versions are displayed to the observer together with the original object. Then he is asked to provide a score for each object, reflecting the degree of perceived visual similarity, between four (identical to the original) and zero (worst case). The objects are displayed during about three minutes and interactions are allowed (rotation, scaling, translation). It is important to note that since the observer can see all the six degraded versions on the same screen, there is no need to establish a referential range, since he will naturally put a "0" for the object he finds the most degraded and "4" to the best one. In order to avoid the effect of the spatial and temporal sequencing factors, the sequence of four models and six degraded versions are randomly generated for each participant. The mean opinion score (MOS) is then computed for each object of the corpus:

$$MOS_i = \frac{1}{n} \sum_{j=1}^n m_{ij}. \quad (11)$$

$MOS_i$  is the mean opinion score of the  $i^{th}$  object,  $n$  is the number of test subjects, and  $m_{ij}$  is the score ( $\in [0, 4]$ ) given by the  $j^{th}$  subject to the  $i^{th}$  object.

This subjective experiment has been conducted on 11 researchers (students and staff) from the LIRIS lab, INSA-Lyon. Firstly, in order to validate the relevance of the results, we have checked if there exist significant rating distribution differences from subject to subject in the subjective scores; if differences exist, that would suggest that the assessment protocol has not been consistently understood by the subjects. For that purpose, we have conducted a one-way Analysis of Variance (see Table II). The hypothesis of significant rating variation between subjects (with a standard confidence  $\alpha = 5\%$ ) is clearly rejected ( $p = 37\% > \alpha$ ). That means that our experiment has been correctly designed and therefore the scores do not need normalization.

The mean opinion scores are presented for every object in Table III together with their standard deviations, and summarized on Figure 13. Firstly, it appears that the mean opinion score behaviors remain around the same whatever the object which is considered; that somehow demonstrates the consistency of our measure even for models from different natures or appearances. Secondly, we can

Table III. Mean Opinion Scores (MOS) and their Standard Deviation ( $\sigma$ ) for Different Objects, for Different Noise Strengths: Low, Medium, and High and at Different Locations: Rough and Smooth Parts (HR=“High Noise on Rough parts”, HS=“High Noise on Smooth Parts”, MR=“Medium Noise on Rough Parts” etc.)

Object		HR	HS	MR	MS	LR	LS
<b>Armadillo</b>	MOS	2.00	0.18	2.82	1.14	3.36	2.45
	$\sigma$	1.09	0.40	0.98	0.78	0.81	1.04
<b>Bimba</b>	MOS	2.73	0.00	3.09	0.82	3.36	1.91
	$\sigma$	0.78	0.00	0.70	0.40	0.67	0.83
<b>Dyno</b>	MOS	1.55	0.09	3.09	1.55	3.36	2.18
	$\sigma$	0.82	0.30	0.83	0.82	0.81	0.98
<b>Lion Head</b>	MOS	2.72	0.00	3.27	0.82	3.45	2.18
	$\sigma$	0.79	0.00	0.65	0.60	0.52	0.75
<b>The whole corpus</b>	MOS	2.25	0.07	3.07	1.08	3.39	2.18
	$\sigma$	0.99	0.25	0.90	0.71	0.69	0.90

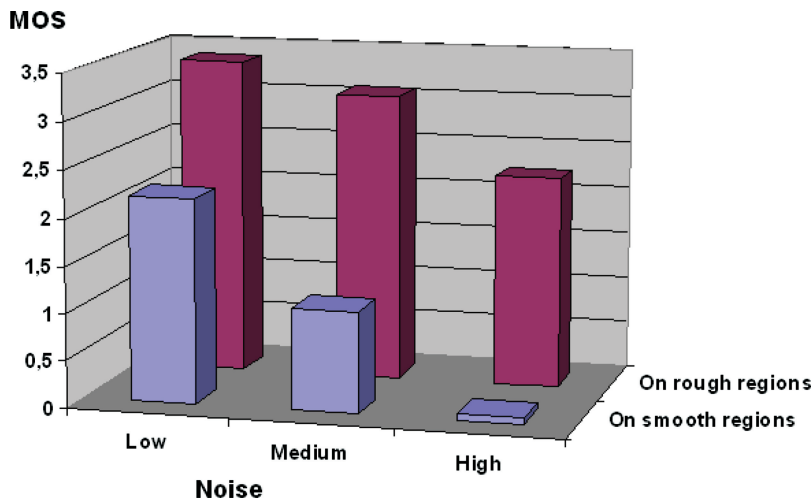


Fig. 13. Average Mean Opinion Scores (MOS) for different noise strengths at different locations.

observe that the MOS (i.e. the visual similarity) is much larger when the noise is applied on rough regions. That clearly attests that our roughness measure successfully detects high masking regions. In particular, the MOS (for the whole corpus) corresponding to the lowest noise on smooth regions (2.18) is lower than those corresponding to the highest noise applied on rough regions (2.25). Accordingly, the rough regions detected by our algorithm are able to efficiently hide large geometric distortions, thanks to the masking phenomenon.

The standard deviations ( $\sigma$ ) appear quite high since only a limited number of integer values were available for rating (from zero to four). However, it is interesting to observe that for some models (Armadillo and Dyno) the standard deviation is significantly higher than for others, whatever the version to rate. That illustrates the fact that some geometrical features are more difficult to rate than others, indeed they may represent different visual qualities or visual appearances depending on the user cognitive behavior.

## 5. APPLICATIONS

We provide here illustrations of utilization of our roughness measure within compression and watermarking applications. These two examples illustrate how the roughness analysis can improve such



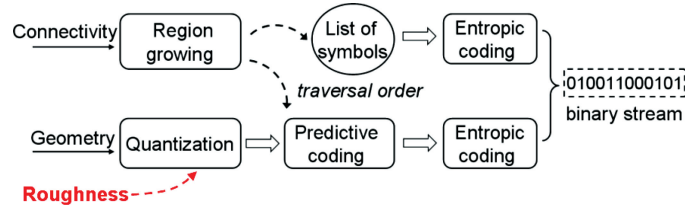


Fig. 14. Generic scheme of single rate compression algorithms. The roughness analysis can be introduced within the quantization step to improve the compression rate.

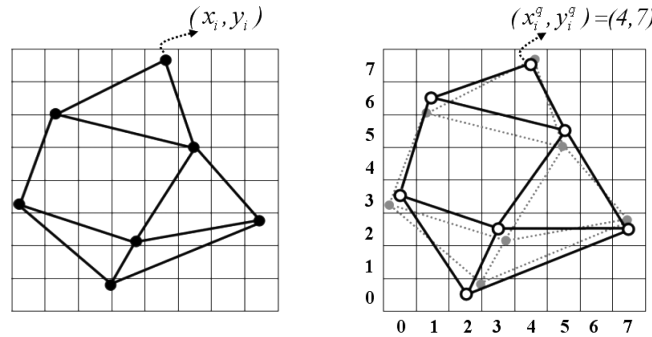


Fig. 15. Geometric uniform quantization of a polygonal mesh on a  $8 \times 8$  grid (3-bits quantization).

applications through the masking effect. Of course our objective is not to overcome every existing compression and watermarking algorithms, but rather to demonstrate how the roughness can be integrated into common techniques from the state-of-the-art so as to improve their results.

## 5.1 Compression

The main goal of a compression method is to find an optimal compromise between the bitrate and the distortion: how to obtain a minimal file size while preserving a high visual quality. Most of the existing single rate compression algorithms for 3D mesh follow the same scheme (see figure 14): the geometry and the connectivity are encoded separately but not independently. The connectivity is encoded by a region growing approach which provides a list of symbols and defines a traversal order of the mesh while the geometry is coded by these three main steps:

- (1) quantization, that transforms floating values into integers by projecting the data onto a 3D uniform grid for instance,
- (2) prediction guided by the traversal order, in order to reduce the redundancy of the information,
- (3) entropic coding, that yields to reduce the average number of bits allocated to each quantized value.

The only step among them which modifies the data in an irreversible way is the quantization. Figure 15 illustrates a 3-bits quantization of the geometric coordinates of a 2D polygonal mesh: coordinates  $(x_i, y_i)$  of each vertex are quantized on a  $8 \times 8$  grid, and are thus represented by integers  $(x_i^q, y_i^q)$  with  $x_i^q$  and  $y_i^q \in [0, 7]$ . Their quantized coordinates can now be encoded with 3 bits (instead of 32 for a floating value). New positions of the vertices, after decoding, are modified: they are the centers of the square cells containing them. Whereas current compression techniques consider a same quantization level for the whole object, our objective is to consider different levels for different parts of the object. Since rough parts can hide higher geometric distortion, it seems relevant to lower the quantization level for such parts so as to improve the compression rate while keeping a good visual quality. Chow [1997] has

considered a similar *variable quantization* idea: the mesh is segmented into several spatial regions according to curvature or triangle size, and then a different quantization step is chosen for each region. However their criteria (curvature, triangle size) are not really related to masking.

Our approach is the following: the roughness is classified into two clusters, *rough* and *smooth*, using the recent Markovian clustering described in [Lavoué and Wolf 2008] which improves the standard K-means algorithm by allowing us to obtain spatially coherent clusters (see Figure 16 top left). Then the connectivity is encoded using the *Face Fixer* algorithm [Isenburg and Snoeyink 2001]. This creates a traversal order of the mesh which is used to process a simple differential geometry coding: a vertex is encoded by its difference vector to the previous one. The differential coordinates are then quantized according to the corresponding cluster labels, a *rough* vertex is associated with a lower quantization level than a smooth one. Finally these quantized values are encoded using arithmetic coding; since several vertices are associated with a lower precision, the distribution of all the values for the whole mesh is thus associated with a lower entropy, and then the arithmetic coding provides better results than for a classical *static* quantization.

We have tested our algorithm on the Lion Head model (see figure 16). The connectivity has been encoded on 13KB using the Face Fixer algorithm [Isenburg and Snoeyink 2001] while the geometry has been encoded according to three different approaches:

- static quantization, 11 bits (bottom left of the figure),
- static quantization, 10 bits (bottom right of the figure),
- variable quantization, 11-9 bits (top right of the figure): *rough* parts (green) of the Lion Head have been quantized on 9 bits, while smooth parts (blue) have been quantized on 11 bits.

After our variable quantization, the entropic coding yields to 70KB for the geometry, against, respectively, 84KB and 66KB for 11-bits and 10-bits quantizations. Figure 16 shows clearly that after decompression the resulting 3D model from our variable quantization is visually very close to the 11-bits quantized one, since the distortions are *hidden* in the rough parts. However, the compression rate is about 17% better. If we compare our result to the 10-bits quantization, both methods produce approximately the same compressed size (70KB against 66KB for geometry), but the visual quality is much better for our algorithm.

It is important to note that the gain in compression that our algorithm provides regarding static quantization, depends on the amount of rough regions that the 3D mesh contains.

## 5.2 Watermarking

Watermarking consists in inserting/hiding information (i.e. the *mark*, usually a binary word) into a digital signal. For 3D mesh, the mark is inserted by slightly modifying the coordinates of the vertices according to a given strength. For copyright protection, the watermark has to be *robust* to survive (i.e. remain detectable) through malicious attacks (geometry or connectivity modifications); hence, one of the main issues is to optimize the trade-off between imperceptibility and robustness. In video and 2D image watermarking, a lot of work has been done to integrate some perceptual aspects in order to optimize this trade-off [Wolfgang et al. 1999], however, that is not the case for 3D mesh, except some recent works [Uccheddu 2007].

Our objective is to drive the strength of the watermark according to the roughness of the associated region: a rough region can easily *hide* a geometric distortion and thus can be more strongly watermarked than a smooth one; hence, the overall robustness of the mark should probably be increased, while keeping a correct visual quality. Among existing mesh watermarking techniques, Ohbuchi et al. [2002] propose a framework based on spectral analysis: the mesh is firstly segmented arbitrarily into several regions and each region is decomposed on the eigenvectors of its Laplacian matrix, which reflects a

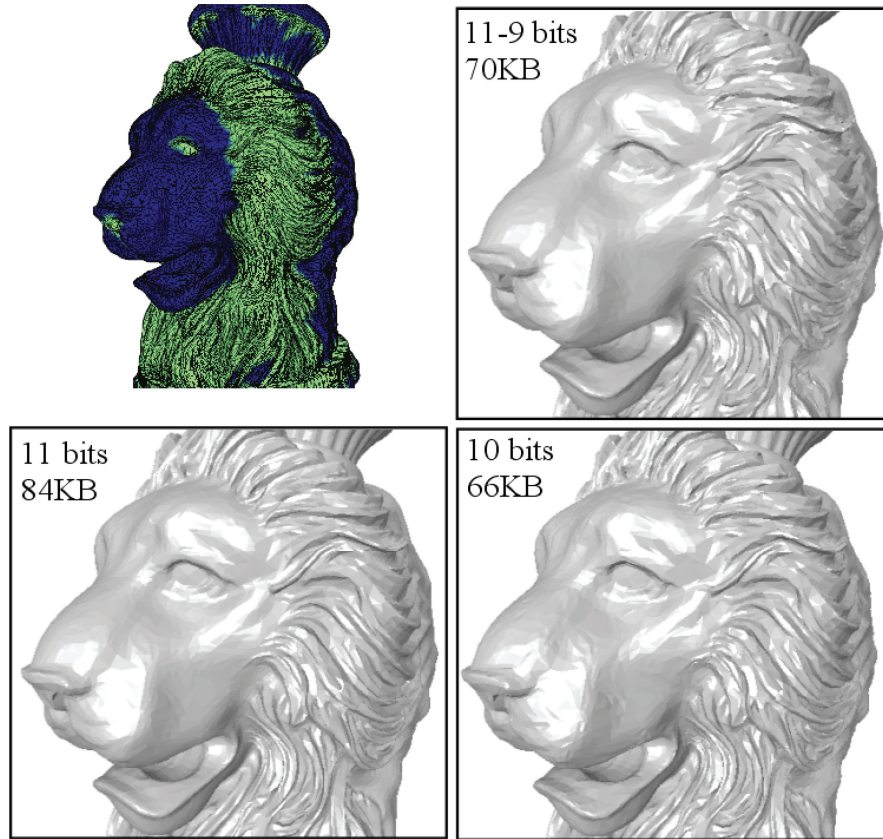


Fig. 16. *Top*: Clustering results and compression results with our roughness based scheme (9 bits quantization for rough regions and 11 bits for smooth ones). *Bottom*: Compression results with static 11 bits and 10 bits quantization. The size of the encoded geometry is displayed on the figures.

kind of Fourier spectral decomposition. Then the spectral coefficients  $c_i$  are modified according to a modulating vector  $V = (v_1, v_2, \dots, v_m)$ ,  $v_i \in \{-1, 1\}$  which represents the mark to insert:

$$\hat{c}_i = c_i + v_i \cdot \alpha, \quad (12)$$

with  $\hat{c}_i$  the watermarked spectral coefficient,  $c_i$  the original one, and  $\alpha$  the global watermarking strength which controls the energy of the embedded watermark. Finally the mesh is reconstructed according to these modified coefficients. At the extraction the algorithm compares the spectral coefficients from the original and watermarked objects in order to retrieve the binary mark.

In the original algorithm from Ohbuchi et al. [2002], this strength is the same for every region. Our approach involves two modifications of this original algorithm:

- (1) The mesh is no longer segmented arbitrarily, but according to its roughness scalar field. We have adapted the recent algorithm from [Cohen-Steiner et al. 2004] such as to take into account roughness attributes, instead of planarity criteria. Segmentation results are displayed in figure 17.
- (2) The watermarking strength  $\alpha$  is no longer global but depends on the mean roughness value for each region:

$$\alpha_j = \alpha \times (R_j - R_{min}) / (R_{mean} - R_{min}), \quad (13)$$

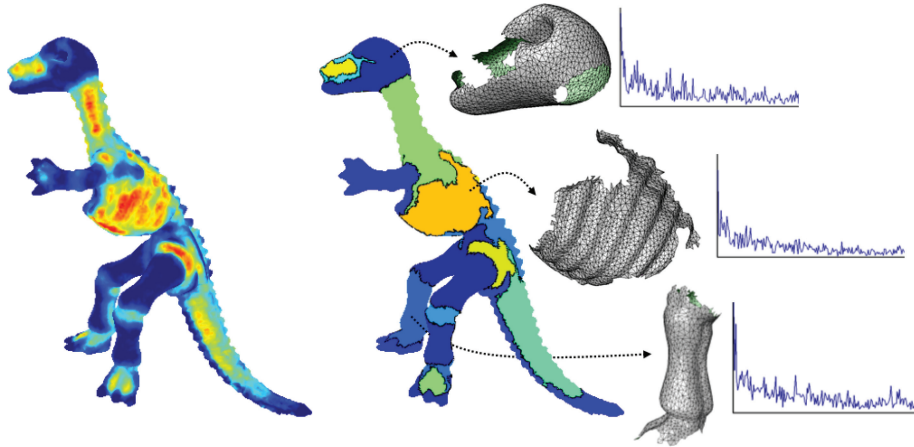


Fig. 17. *Left*: Roughness map of the Dyno mesh. *Right*: Segmentation results, with mean roughness value for each region (with the same color map). Three examples of regions with their Laplacian spectrum.

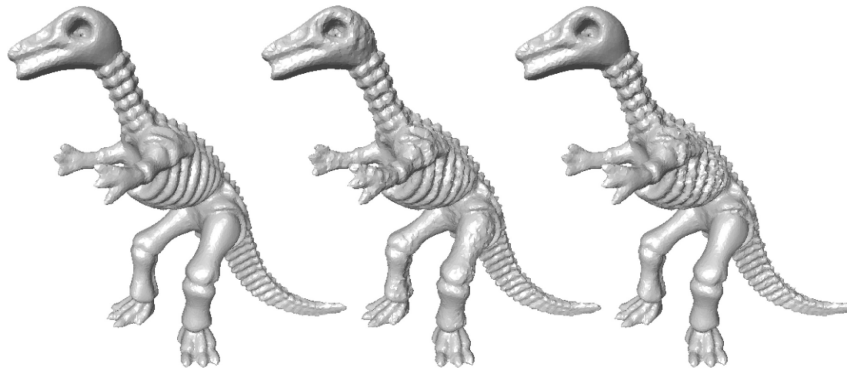


Fig. 18. *Left*: Original Dyno mesh. *Middle*: Watermarked mesh with the classical Ohbuchi approach ( $\alpha = 0.001$ ). *Right*: Watermarked mesh with our roughness based approach ( $\alpha = 0.0008$ ).

where  $R_j$  is the roughness of region  $j$ , and  $R_{min}$  and  $R_{mean}$ , respectively, the minimum and mean roughness over the whole mesh. This simple algorithm provides very good results: a smooth region is associated with a very weak watermark strength while a rougher region is more strongly modified.

Figure 18 illustrates the visual appearance of the Dyno mesh after watermarking using respectively the classical algorithm with a constant strength ( $\alpha = 0.001$ ) and our roughness based approach ( $\alpha = 0.0008$ ). When our approach is used, the visual quality of the watermarked mesh appears better, since the distortion is concentrated on very rough parts; hence, the smooth regions like the head or the legs of the Dyno are preserved.

In order to properly compare both approaches we have made robustness experiments using these parameters ( $\alpha = 0.001$  for the classical one and  $\alpha = 0.0008$  for our method): we compute the correlation between inserted and extracted marks, after two geometric attacks (i.e. modifications): noise addition and nonuniform scaling. In the following results, for each correlation value, we have repeated 50 times the insertion, the attack, and the extraction, with random bit patterns of length 64 bits and then averaged the obtained correlations. Results are the following:

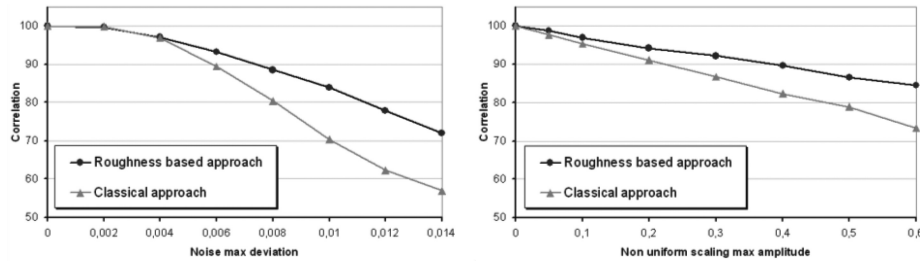


Fig. 19. Correlation (%) of the extracted watermark with the inserted one, under noise addition and non-uniform scaling attacks.

**Noise addition:** We modify the three coordinates of each vertex of the mesh, according to a randomly chosen offset between *zero* and a maximum deviation  $E_{max}$ . Figure 19 shows the average correlation of the extracted marks for both methods, after noise addition attack, according to increasing  $E_{max}$  values. Our approach significantly improves the robustness; for instance, our correlation reaches 84% for a noise with amplitude 0.01 against 70% for the classical approach.

**Nonuniform scaling:** For each axes (X, Y, and Z), we compute a scaling value, randomly chosen between  $1 - S_{max}$  and  $1 + S_{max}$  and we multiply the corresponding coordinates by this value. Figure 19 shows the extracted average correlation, according to increasing  $S_{max}$  values. Like for the noise addition attack, our scheme gives much better results than the classical one.

As for the compression application, the roughness analysis has improved the efficiency of the watermarking: the robustness regarding attacks has been increased while the visual quality of the watermarked mesh is better.

## 6. 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 extracts rough regions while not misclassifying *edge* parts and smooth parts. Extensive comparisons with existing algorithms have demonstrated the superiority of our approach regarding robustness and accuracy. Moreover, a subjective evaluation has been conducted which clearly demonstrates the visual masking capacity of rough regions detected by our estimator.

Finally, two applications to compression and watermarking have been presented; they illustrate the fact that our measure can be easily and advantageously integrated to common processing operations to improve the results by hiding the artifacts into high masking regions.

The present work constitutes a first step toward the complete understanding of 3D geometric masking; indeed, we have demonstrated that a rough region, calculated at a given scale, is able to efficiently mask geometric artifacts, however, the quantitative evaluation of the masking effect involves a precise knowledge of the frequency distribution (both the magnitude and orientation) of the masker and the signal to be masked.

## ACKNOWLEDGMENTS

We would like to thank Elisa Drelie Gelasca and Massimiliano Corsini for the source code of their roughness estimators. We also thank the anonymous reviewers for helping us to improve this paper.

This work is partially supported by the French National Research Agency (ANR) through MADRAS project (ANR-07-MDCO-015).

## 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.
- BOLIN, M. AND MEYER, G. 1998. A perceptually based adaptive sampling algorithm. In *ACM Siggraph*. 299–309.
- CAMPBELL, F. AND KULIKOWSKI, J. 1966. Orientation selectivity of the human visual system. *Journal of Physiology* 187, 437–445.
- CCIR. 1986. Method for the subjective assessment of the quality of television pictures, recommendation 500-3. In *Recommendations and reports of the CCIR, International Telecommunication Union*, Geneva, Switzerland.
- CHOW, M. 1997. Optimized geometry compression for realtime rendering. In *IEEE Visualization*. 347–354.
- COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. In *ACM Siggraph*. 905–914.
- COHEN-STEINER, D. AND MORVAN, J. 2003. Restricted delaunay triangulations and normal cycle. In *19th Annu. ACM Sympos. Comput. Geom.*
- CORSINI, M., DRELIE GELASCA, E., EBRAHIMI, T., AND BARNI, M. 2007. Watermarked 3D mesh quality assessment. *IEEE Transaction on Multimedia* 9, 2 (February), 247–256.
- DALY, S. 1993. *The visible differences predictor: An algorithm for the assessment of image fidelity*. MIT Press, Cambridge, MA, 179–206.
- DUMONT, R., PELLACINI, F., AND FERWERDA, J. 2003. Perceptually-driven decision theory for interactive realistic rendering. *ACM Transactions on Graphics* 22, 2, 152–181.
- ECKERT, M. AND BRADLEY, A. 1998. Perceptual quality metrics applied to still image compression. *Signal Processing* 70, 3, 177–200.
- FERWERDA, J., PATTANAIK, S., SHIRLEY, P., AND GREENBERG, D. 1997. A model of visual masking for computer graphics. In *ACM Siggraph*. 143–152.
- HARMON, L. AND JULESZ, B. 1973. Masking in visual recognition: effects of two-dimensional filtered noise. *Science* 180, 1194–1197.
- 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.
- ISENBURG, M. AND SNOEYINK, J. 2001. Face fixer: Compressing polygon meshes with properties. In *ACM Siggraph*. 263–270.
- JONES, T. R., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. In *SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers*. ACM, New York, 943–949.
- 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.
- LAVOUÉ, G. AND WOLF, C. 2008. Markov random fields for improving 3D mesh analysis and segmentation. In *Eurographics Workshop on 3D Object Retrieval*. 25–32.
- 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.
- OHBUCHI, R., MUKAIYAMA, A., AND TAKAHASHI, S. 2002. A frequency-domain approach to watermarking 3D shapes. *Computer Graphic Forum* 21, 3, 373–382.
- PAN, Y., CHENG, I., AND BASU, A. 2005. Quality metric for approximating subjective evaluation of 3-d objects. *IEEE Transactions on Multimedia* 7, 2, 269–279.
- QU, L. AND MEYER, G. W. 2006. Perceptually driven interactive geometry remeshing. In *I3D ’06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*. ACM, New York, 199–206.
- REDDY, M. 2001. Perceptually optimized 3D graphics. *IEEE Computer Graphics and Applications* 21, 5, 68–75.
- ROGOWITZ, B. E. AND RUSHMEIER, H. E. 2001. Are image quality metrics adequate to evaluate the quality of geometric objects? *SPIE Human Vision and Electronic Imaging VI 4299*, 1, 340–348.
- 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.
- TIAN, D. AND ALREGIB, G. 2004. Fqm: A fast quality measure for efficient transmission of textured 3D models. In *ACM Multimedia*. 684–691.
- UCCHEDDU, F. 2007. Robust and imperceptible watermarking of 3D meshes. Ph.D. thesis, University of Florence, Italy.

- WANG, K., LAVOUÉ, G., DENIS, F., AND BASKURT, A. 2007. Hierarchical blind watermarking of 3D triangular meshes. In *IEEE International Conference on Multimedia and Expo*. 1235–1238.
- WANG, Z., BOVIK, A., SHEIKH, H., AND SIMONCELLI, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4, 1–14.
- WATSON, B., FRIEDMAN, A., AND MCGAFFEY, A. 2001. Measuring and predicting visual fidelity. In *ACM Siggraph*. 213–220.
- WILLIAMS, N., LUEBKE, D., COHEN, J., KELLEY, M., AND SCHUBERT, B. 2003. Perceptually guided simplification of lit, textured meshes. In *ACM Symposium on Interactive 3D Graphics*. 113–121.
- WOLFGANG, R., PODILCHUK, C., AND DELP, E. JUL 1999. Perceptual watermarks for digital images and video. *Proceedings of the IEEE* 87, 7, 1108–1126.
- 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.

Received October 2007; July 2008, September 2008; accepted October 2008