# A Benchmark for 3D Mesh Watermarking

Kai Wang[1], Guillaume Lavoué[1], Florence Denis[2], Atilla Baskurt[1], and Xiyan He[3]
[1,2] Université de Lyon, CNRS
[1] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[2] Université Lyon 1, LIRIS, UMR5205, F-69622, France
[3] Institut Charles Delaunay (FRE 2848 CNRS), Université de Technologie de Troyes, F-10010, France

***Abstract*—This paper presents a benchmarking system for the evaluation of robust mesh watermarking methods. The proposed benchmark has three different components: a "standard" mesh model collection, a software tool and two application-oriented evaluation protocols. The software tool integrates both geometric and perceptual measurements of the distortion induced by watermark embedding, and also the implementation of a variety of attacks on watermarked meshes. The two evaluation protocols define the main steps to follow when conducting the evaluation experiments. The efficiency of the benchmark is demonstrated through the evaluation and comparison of two recent robust algorithms.**

***Keywords*—Mesh watermarking, benchmark, distortion, attack, robustness**

## 1. INTRODUCTION

With the increasing use of mesh models (particularly on Internet) during the last decade, it is now essential to establish an efficient mechanism for their copyright protection. Robust watermarking seems a good solution to this emerging problem. The basic idea is to embed a piece of copyright-related information (*i.e.* the watermark) into the functional part of a mesh file. The embedded watermark should be robust against various attacks on the watermarked model and also be imperceptible to human eyes. So far, a number of robust algorithms have been proposed; readers could refer to [1] for a comprehensive survey on the relevant state of the art.

In this paper, we focus on the evaluation of robust mesh watermarking schemes. Indeed, when a new scheme is proposed, we often want to compare it with some existing methods so as to fairly assess its strong and weak points. However, at present, it seems difficult and time-consuming to carry out such a comparison, mainly because the authors of different methods often use different mesh models, distortion metrics, attacks and evaluation methodologies when reporting their experimental results. Therefore, it is necessary to propose a benchmarking system to the mesh watermarking community, with the objective to attain a fair and easy comparison between different algorithms.

Several benchmarking systems have been constructed for evaluating image watermarks, such as Stirmark [2], Checkmark [3] and Optimark [4]. In contrast, to the best of our knowledge, the benchmarking of 3D mesh watermarks was addressed only by Bennour and Dugelay [5]. They propose to use some existing software packages to measure the geometric distance between original and watermarked models, and to conduct attacks on watermarked meshes. The authors also propose to use a four-element structure to report the overall performance of a robust scheme. Compared to their proposal, our contributions are threefold: 1) We provide a publicly available data set of mesh models and an open-source software tool for the evaluation of robust mesh watermarks[1]. The provided software contains a number of typical attacks, a perceptual distortion metric and the legacy implementation of several largely used geometric distortion measurements. 2) Two protocols are established for the payload-distortion-robustness evaluation, and they define de main steps to follow when conducting the experimental assessment. The protocols, along with the mesh data set and the software tool, constitute a well-structured testing framework for robust mesh watermarking schemes. 3) Two recent algorithms are compared by using the proposed benchmark. The procedure of this comparison demonstrates that our evaluation framework is easy to use and also very effective.

The remainder of this paper is organized as follows: Section 2 introduces the evaluation targets of the proposed benchmark; Sections 3 and 4 present respectively the distortion metrics and the attacks integrated in our software tool; we propose two different application-oriented evaluation protocols in Section 5; the comparison results of two state-of-the-art methods are presented in Section 6; finally, we draw conclusion in Section 7.

## 2. EVALUATION TARGETS

A robust watermarking scheme is often evaluated in four different aspects: *payload*, *distortion*, *robustness* and *security*. The *payload* is the number of bits of the hidden message conveyed by the watermark. The *distortion* measures the difference between the original

This work is in part supported by China Scholarship Council of the Chinese government and French National Research Agency (ANR) through COSINUS program (project COLLAVIZ n°ANR-08-COSI-003).

[1]Available at http://liris.cnrs.fr/meshbenchmark/

cover content and its watermarked version. The *robustness* indicates how resistant the watermark is against various routine operations on the watermarked content. A *secure* watermarking scheme should be able to withstand the malicious attacks that aim to break down the whole watermarking-based copyright protection system through, for instance, secret key disclosure or inversion of the watermark embedding procedure. In the proposed benchmark, we consider only the payload, distortion and robustness evaluations, while discarding the security metric. The main reason is that the research on mesh watermarking is still in its early stage [1] and until now the community has been interested in achieving robustness against connectivity attacks (*e.g.* surface simplification and remeshing) while paying little attention on security, a rather high-level requirement. Finally, when reporting the evaluation results, the authors should also indicate whether their scheme is blind, semi-blind or non-blind.

When evaluating a robust mesh watermarking scheme by using the above metrics, we also need a well-defined protocol which indicates the steps to follow when conducting the experiments. Before presenting our application-oriented evaluation protocols in Section 5, we will first explain how we measure the distortion induced by watermark embedding and the various attacks against which we would like to test the robustness.

## 3. Distortion Metrics

The watermark embedding process introduces some amount of distortion to the original cover mesh. This distortion can be measured *geometrically* or *perceptually*. For the geometric measurement, we propose to use the maximum root mean square error (MRMS). In general, the root mean square error (RMS) from one 3D surface $S$ to another 3D surface $S^{'}$ is defined as:

$$d_{RMS}(S,S^{'}) = \sqrt{\frac{1}{|S|}\int\int_{p\in S} d\left(p,S^{'}\right)^2 dS}, \tag{1}$$

where $p$ is a point on surface $S$, $|S|$ is the area of $S$, and $d(p,S^{'})$ denotes the point-to-surface distance between $p$ and $S^{'}$. This RMS distance is not symmetric and generally we have $d_{RMS}(S,S^{'}) \neq d_{RMS}(S^{'},S)$. Therefore, we can define the MRMS distance between a cover mesh $\mathcal{M}$ and its watermarked version $\mathcal{M}^{'}$ as:

$$d_{MRMS}(\mathcal{M},\mathcal{M}^{'}) = \max\left(d_{RMS}(\mathcal{M},\mathcal{M}^{'}), d_{RMS}(\mathcal{M}',\mathcal{M})\right). \tag{2}$$

Different from the simple vertex-to-vertex distance metrics (*e.g.* the vertex coordinates PSNR), MRMS measures the surface-to-surface distance between two meshes. The distortion measured by MRMS is more accurate, especially when the two meshes under comparison do not have the same connectivity configuration. We have included the legacy MRMS implementation of Metro [6] in our benchmarking software.

It is well known that the geometric surface-to-surface distances, such as MRMS, do not correctly reflect the visual difference between two meshes [7]. Thus, we need another perceptual metric to measure the visual distortion induced by watermark embedding. For this purpose, we have considered the mesh structural distortion measure (MSDM) proposed by Lavoué *et al.* [7], and have integrated it in the benchmarking software. This metric follows the concept of structural similarity recently introduced by Wang *et al.* [8] for 2D image quality assessment, and well reflects the perceptual distance between two 3D objects. The local MSDM distance between two mesh local windows $p$ and $q$ (respectively in $\mathcal{M}$ and $\mathcal{M}^{'}$) is defined as follows:

$$d_{LMSDM}(p,q) = (0.4\times L(p,q)^3 + 0.4\times C(p,q)^3 + 0.2\times S(p,q)^3)^{\frac{1}{3}}, \tag{3}$$

where $L$, $C$ and $S$ represent respectively curvature, contrast and structure comparison functions (please refer to [7] for more details). The global MSDM distance between two meshes $\mathcal{M}$ and $\mathcal{M}^{'}$ (both having $n$ vertices), is defined by a Minkowski sum of the meshes' $n$ local MSDM distances measured on their $n$ vertices:

$$d_{MSDM}(\mathcal{M},\mathcal{M}^{'}) = \left(\frac{1}{n}\sum_{i=1}^{n} d_{LMSDM}(p_i,q_i)^3\right)^{\frac{1}{3}} \in [0,1). \tag{4}$$

Its value tends toward 1 (theoretical limit) when the measured objects are visually very different and is equal to 0 for identical objects. The main reasons for choosing this perceptual distortion metric are its strong robustness and its high correlation with the subjective evaluation results given by human beings [7].

## 4. Attacks

In general, there are three kinds of routine attacks on a watermarked mesh: *file attack*, *geometry attack* and *connectivity attack*. In the following, we will give examples for each kind of attack and present the corresponding implementations in our benchmarking software.

### 4.1 File attack

This attack reorders the vertices and/or the facets in the mesh file, and does not introduce any modification to the mesh shape. A robust mesh watermark should be perfectly invariant to this kind of attack. When carrying out the file attack, the benchmarking software uses a randomly selected key to rearrange the vertex and facet indices in their corresponding lists in the mesh file.

### 4.2 Geometry attack

In a geometry attack, only the vertex coordinates are modified while the mesh connectivity (*i.e.* the adjacency relationship between vertices) is kept unchanged. Our benchmarking software integrates the implementation of the following geometry attacks.

**Similarity transformation**. This operation includes translation, rotation, uniform scaling and their combination. Like the above vertex/facet reordering operation, the similarity transformation always keeps the mesh shape intact. Actually, these two kinds of operations are jointly called content-preserving attacks, through which a robust watermark, or even a fragile watermark, should

be able to survive. In our implementation, in each run of the similarity transformation, the watermarked mesh is successively subject to a random translation, a random rotation and a random uniform scaling.

**Noise addition**. This attack aims to simulate the artifacts introduced during mesh generation and the errors induced during data transmission. We propose to add pseudo-random noises on vertex coordinates $x_i$ according to the following equation (resp. $y_i$, $z_i$):

$$x_i^{'} = x_i + a_i.\bar{d}, \tag{5}$$

where $\bar{d}$ denotes the average distance from vertices to object center, and $a_i$ is the noise strength for $x_i$. The object center is calculated by using the analytic and continuous volume moments of the mesh [9], which is much more robust than simply calculating it as the average position of the mesh vertices [10]. $a_i$ is a pseudo-random number uniformly distributed in interval $[-A, A]$, with $A$ the maximum noise strength. Figure 1.(b) illustrates a noised Bunny model ($A = 0.30\%$).

**Smoothing**. Surface smoothing is a common operation used to remove the noise introduced during the mesh generation process through 3D scanning. For mesh watermark benchmarking, we choose to carry out Laplacian smoothing [11] on watermarked meshes, with different iteration numbers $N_{itr}$ while fixing the deformation factor $\lambda$ as $0.10$. Figure 1.(c) shows a smoothed Bunny model ($\lambda = 0.10, N_{itr} = 30$).

**Vertex coordinates quantization**. This operation is largely used in mesh compression. Under a $R$-bit uniform quantization, the $x$ (resp. $y$, $z$) coordinate of each vertex is rounded to one of the $2^R$ quantized levels.

### *4.3 Connectivity attack*

In a connectivity attack, the mesh connectivity information, *i.e.* the adjacency relationship between vertices, is changed. Meanwhile, the coordinates of the original vertices may also be modified. We have implemented the following connectivity attacks in the software tool.

**Simplification**. The original version of a mesh model (especially those obtained by a 3D scanning) usually has a very high complexity, sometimes with more than 1 million vertices. This high complexity is necessary to ensure a good precision. In practical applications, the watermark is often embedded in the original complex model, and then the mesh is simplified so as to adapt to the capacity of the available resources. In the benchmarking software we integrated the mesh simplification algorithm of Lindstrom and Turk [12], which provides a good trade-off between the precision of the simplified model and the computational efficiency. The user can designate the edge reduction ratios $E_{sim}$ of the simplification operations. Figure 1.(d) shows a simplified Bunny model ($E_{sim} = 95\%$).

**Subdivision**. In this operation, vertices and edges are added to the original mesh to obtain a modified version that is normally smoother and of a higher visual quality. We suggest to test the watermark robustness against three typical subdivision schemes, always with one iteration: the simple midpoint scheme, the $\sqrt{3}$ scheme and the Loop scheme [13].

**Cropping**. In this attack, one part of the watermarked mesh is cut off and thus lost. This attack happens when we create a new model by combining parts extracted from several other objects. We propose to conduct the copping attacks with different approximative vertex cropping ratios $V_{cr}$. In our implementation, for each cropping ratio, 3 attacked models are generated. These models are obtained by cropping the original stego mesh along 3 randomly selected orthogonal axes. Figure 1.(e) illustrates a cropped Bunny model ($V_{cr} = 10\%$).

Finally, it is worth pointing out that it is important to repeat the attacks with a random nature (*i.e.* file attack, similarity transformation, noise addition and cropping), for at least 3 times, in order to ensure the reliability of the obtained robustness evaluation results.

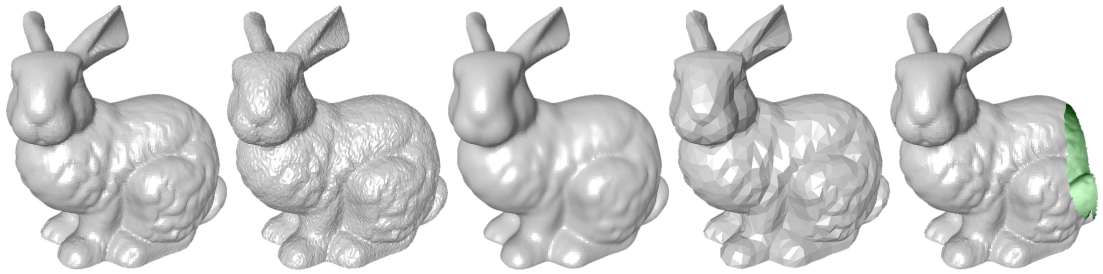
## 5. Evaluation Protocols

The objective of a watermark evaluation protocol is to define the main steps to follow when conducting the experimental assessment of a watermarking scheme. In the case of image watermarking, the authors of Stirmark [2] propose to first fix the watermark payload at about 70 bits and also to limit the induced distortion to be higher than 38 dB in terms of PSNR. After that, Stirmark system carries out a series of attacks on the watermarked image. Then, the user tries to extract watermarks from the obtained attacked stego images. Finally, several plots or tables are reported, which indicate the robustness metric (*e.g.* bit error rate of the extracted watermark) versus the amplitudes of the different kinds of attacks.

We define here two similar protocols for the evaluation of robust mesh watermarking schemes. We call the first protocol *perceptual-quality-oriented* and the second one *geometric-quality-oriented*. The motivation for establishing two different protocols is that different mesh-based applications have very different restrictions on the geometric and the perceptual distortions induced by watermark embedding. For example, for the meshes used in digital entertainment, we should first of all ensure that the induced distortion is not annoying to human eyes (*i.e.* the watermarked model should have a very high visual quality), while the amount of induced geometric distortion is less important. On the contrary, for the meshes used in computer-aided design and medical imaging, it is often required that the geometric distortion should be very small, while the visual quality of the watermarked model is relatively less important.

The perceptual-quality-oriented evaluation protocol consists of the following steps:

1) Embed a watermark $W$ in a test mesh $\mathcal{M}$ by using a secret key $K$ to obtain a watermarked model $\mathcal{M}^{'}$; make sure that the induced perceptual distortion $d_{MSDM} \leq 0.20$ and the induced geometric distortion $d_{MRMS} \leq 0.08\%.l_{bbd}$, where $l_{bbd}$ denotes the diagonal length of the mesh's bounding box.



(a) (b) (c) (d) (e)

Fig. 1. The Bunny model and four attacked versions: (a) the original mesh with 34835 vertices and 104499 edges; (b) after noise addition ($A = 0.30\%$); (c) after Laplacian smoothing ($\lambda = 0.10, N_{itr} = 30$); (d) after simplification ($E_{sim} = 95\%$); (e) after cropping ($V_{cr} = 10\%$).

TABLE 1
ATTACKS USED IN THE EVALUATION PROTOCOLS.

| Attack | Parameter | Parameter values |
|---|---|---|
| File attack | times | 3 |
| Similarity transformation | times | 3 |
| Noise addition[a] | $A$ | 0.05%, 0.10%, 0.30%, 0.50% |
| Smoothing ($\lambda = 0.10$) | $N_{itr}$ | 5, 10, 30, 50 |
| Quantization | $R$ | 11, 10, 9, 8, 7 |
| Simplification | $E_{sim}$ | 10%, 30%, 50%, 70%, 90%, 95%, 97.5%[b] |
| Subdivision (1 iteration) | scheme | midpoint, $\sqrt{3}$, Loop |
| Cropping | $V_{cr}$ | 10%, 30%, 50% |

[a] For each noise amplitude, it is necessary to repeat 3 times.
[b] The ratio 97.5% is only for large meshes having $\geq$ 100K vertices.

2) Carry out the suggested attacks listed in Table 1 on the stego mesh $\mathcal{M}'$, by using the provided benchmarking software.
3) Try to detect/extract the embedded watermark $W$ from each of the obtained attacked stego models and record the detection/extraction robustness result, *i.e.* the confidence level of the existence of $W$ in the tested model or the bit error rate of the extracted watermark.
4) For detectable schemes, also try to detect a random watermark $\tilde{W}$ from each of the obtained attacked stego models and record the detection algorithm output, *i.e.* the confidence level of the existence of $\tilde{W}$ in the tested model.
5) Repeat steps 1-4 for several times with different randomly selected watermark sequences and keys.
6) Repeat steps 1-5 for each test mesh from the standard data set.

The two distortion thresholds in the above protocol (*i.e.* 0.20 for $d_{MSDM}$ and $0.08\%.l_{bbd}$ for $d_{MRMS}$) ensure that the obtained stego model is of very high visual quality and that the cover mesh is not too much deformed. The geometric-quality-oriented protocol consists of the same steps; the difference is that we have different constraints on the induced geometric and perceptual distortions as follows: $d_{MRMS} \leq 0.02\%.l_{bbd}$ and $d_{MSDM} \leq 0.30$. The constraint on $d_{MRMS}$ guarantees that only a very small amount of geometric distortion is introduced to the cover mesh. The constraint on $d_{MSDM}$ avoids this small-amount distortion (sometimes of high frequency) from degrading too much the visual quality of the watermarked object. We are prepared to adjust these four thresholds according to the feedbacks from the community. Finally, note that the two MSDM thresholds in the protocols correspond to the calculation in which the radius parameter is fixed as 0.005 [7]. This parameter is used to define the local window size for the local MSDM calculation given in Eq. (3).

Both *readable* (multi-bit) and *detectable* (one-bit) watermarking schemes can be tested by using our protocols. For readable schemes, we suggest to repeat the watermark embedding for at least 5 times on each model and report the averages of the watermark extraction bit error rates (BER) under the different attacks. For detectable schemes, it is suggested that for each test model we repeat the watermark embedding for at least 50 times by using different watermark sequences and keys. The receiver operating characteristics (ROC) curves (*i.e.* the curves describing the relationship between the false positive rate and the false negative rate of the watermark detection) under each kind of attack are plotted as the evaluation results. Finally, note that sometimes it is advised to also test the ROC performance (in addition to the BER performance) of a readable watermarking scheme by considering it as a detectable scheme.

When carrying out comparison between different readable schemes, we have to ensure that they have the same payload. It is proposed to set the payload to one of the following values: 16 bits, 32 bits, 64 bits and $\geq 96$ bits. Compared to the Stirmark protocol which suggests to fix the payload at around 70 bits, in our protocols it is acceptable that a readable mesh watermarking scheme has a relatively low payload such as 16 or 32 bits. We have loosened the payload requirement mainly based on the observations that robust mesh watermarking is a challenging task due to many particular difficulties and that the relevant research is still in its early stage [1]. Consequently, it is reasonable that the evaluation protocols for mesh watermarking should be less stringent than those for image, audio and video watermarking.

Finally, concerning the dataset collection, we have selected several representative meshes (with different shape complexities and numbers of vertices, and used in different applications) as test models, and also acquired the permission to post them on our public server. These models are: Bunny (34835 vertices), Venus (100759 vertices), Horse (112642 vertices), Dragon (50000 vertices), Rabbit (70658 vertices), Ramesses (826266 vertices), Cow (2904 vertices), Hand (36619 vertices), Casting (5096 vertices), and Crank (50012 vertices).

TABLE 2
BASELINE EVALUATION RESULTS OF THE TWO METHODS.

| Protocol | Perceptual | | Geometric | |
|---|---|---|---|---|
| Method | Cho's | Wang's | Cho's | Wang's |
| WM payload (bits) | 64 | 64 | 64 | 64 |
| Embedding time (s) | 7.6 | 439.9 | 11.6 | 377.6 |
| Extraction time (s) | $< 1.0$ | 3.3 | $< 1.0$ | 3.5 |
| $d_{MRMS}$ (w.r.t. $l_{bbd}$) | 0.0080% | 0.069% | 0.012% | 0.018% |
| $d_{MSDM}$ | 0.19 | 0.14 | 0.29 | 0.09 |

TABLE 3
ROBUSTNESS COMPARISON BETWEEN THE TWO METHODS.

| Protocol ⇒ | Perceptual | | Geometric | |
|---|---|---|---|---|
| Attack ⇓ | Cho's BER | Wang's BER | Cho's BER | Wang's BER |
| **File attack** | **0** | **0** | **0** | **0** |
| Similarity transformation | 0 | 0 | 0 | 0 |
| Noise $A = 0.05\%$ | 0.01 | 0 | 0 | 0.02 |
| Noise $A = 0.10\%$ | 0.03 | 0.01 | 0.01 | 0.15 |
| Noise $A = 0.30\%$ | 0.13 | 0.08 | 0.10 | 0.29 |
| Noise $A = 0.50\%$ | 0.28 | 0.16 | 0.24 | 0.40 |
| Smoothing $N_{itr} = 5$ | 0.10 | 0 | 0.06 | 0.06 |
| Smoothing $N_{itr} = 10$ | 0.23 | 0.01 | 0.16 | 0.18 |
| Smoothing $N_{itr} = 30$ | 0.38 | 0.07 | 0.34 | 0.39 |
| Smoothing $N_{itr} = 50$ | 0.45 | 0.14 | 0.42 | 0.51 |
| Quantization $R = 11$ | 0 | 0 | 0 | 0.01 |
| Quantization $R = 10$ | 0.04 | 0.01 | 0.02 | 0.17 |
| Quantization $R = 9$ | 0.14 | 0.01 | 0.06 | 0.27 |
| Quantization $R = 8$ | 0.26 | 0.05 | 0.18 | 0.39 |
| Quantization $R = 7$ | 0.46 | 0.17 | 0.41 | 0.53 |
| **Average geometry attacks** | **0.18** | **0.05** | **0.14** | **0.24** |
| Subdivision Midpoint | 0.04 | 0 | 0.02 | 0 |
| Subdivision $\sqrt{3}$ | 0.14 | 0 | 0.09 | 0.01 |
| Subdivision Loop | 0.16 | 0 | 0.09 | 0.01 |
| Simplification $E_{sim} = 10\%$ | 0.01 | 0 | 0 | 0 |
| Simplification $E_{sim} = 30\%$ | 0.05 | 0 | 0.03 | 0 |
| Simplification $E_{sim} = 50\%$ | 0.18 | 0 | 0.07 | 0.02 |
| Simplification $E_{sim} = 70\%$ | 0.33 | 0 | 0.14 | 0.02 |
| Simplification $E_{sim} = 90\%$ | 0.23 | 0.01 | 0.12 | 0.08 |
| Simplification $E_{sim} = 95\%$ | 0.38 | 0.01 | 0.27 | 0.17 |
| Simplification $E_{sim} = 97.5\%$ | 0.47 | 0.05 | 0.42 | 0.32 |
| Cropping $V_{cr} = 10\%$ | 0.50 | 0.51 | 0.50 | 0.51 |
| Cropping $V_{cr} = 30\%$ | 0.53 | 0.49 | 0.51 | 0.48 |
| Cropping $V_{cr} = 50\%$ | 0.51 | 0.49 | 0.52 | 0.49 |
| **Average connectivity attacks** | **0.27** | **0.12** | **0.21** | **0.16** |
| **Average all attacks** | **0.22** | **0.08** | **0.17** | **0.19** |

## 6. COMPARISON OF TWO RECENT ALGORITHMS

In order to test the usability of the proposed benchmark, we have used it to evaluate and compare two recent blind and robust mesh watermarking schemes: the method of Cho *et al.* [14] that is based on modification of the mean value of the histogram of vertex norms, and the method of Wang *et al.* [10] that is based on modification of the mesh local volume moments. Table 2 presents the baseline evaluation results of the two methods on Venus model, under both protocols. The corresponding robustness evaluation results are presented in Table 3, in which we also list some average BER values under different types of attacks. All the results are averages of 5 trials with randomly selected watermark sequences and keys. We can conclude that in general the method of Wang *et al.* is more suitable to be used in applications that require a high visual quality of the watermarked object, while the method of Cho *et al.* is more appropriate for the applications which have strict restriction on the amount of induced geometric distortion. However, in both kinds of applications, if a strong robustness against connectivity attacks is required, then the method of Wang *et al.* seems the better choice.

## 7. CONCLUSION

We proposed a benchmark for the evaluation of robust mesh watermarking schemes. A software tool, which includes both geometric and perceptual distortion metrics, as well as a large number of attacks, has been implemented. Two different application-oriented mesh watermarking evaluation protocols have been established. Two recent robust algorithms were compared within the proposed benchmarking framework. The data set, the protocol configuration file and the source code of the software are publicly available at http://liris.cnrs.fr/meshbenchmark/. We expect receiving feedbacks from the mesh watermarking community, based on which we could further improve the usability of the benchmark.

## ACKNOWLEDGMENTS

We would like to thank Prof. M. Levoy for granting us the permission to post the Bunny and Dragon models on our public server. The Venus, Horse and Rabbit models are the courtesies of the Cyberware Inc., and the Ramesses, Cow, Hand, Casting and Crank models are the courtesies of the AIM@SHAPE project. We are grateful to Dr. P. Cignoni for allowing us to integrate the Metro tool in our benchmarking software.

## REFERENCES

[1] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Trans. on Multimedia*, vol. 10, no. 8, pp. 1513–1527, 2008.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Proc. of the Int. Workshop on Information Hiding*, 1998, pp. 218–238.

[3] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, and T. Pun, "Second generation benchmarking and application oriented evaluation," in *Proc. of the Int. Workshop on Information Hiding*, 2001, pp. 340–353.

[4] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, and I. Pitas, "A benchmarking protocol for watermarking methods," in *Proc. of the IEEE Int. Conf. on Image Process.*, 2001, vol. 3, pp. 1023–1026.

[5] J. Bennour and J.-L. Dugelay, "Toward a 3D watermarking benchmark," in *Proc. of the IEEE Int. Workshop on Multimedia Signal Process.*, 2007, pp. 369–372.

[6] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Comput. Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.

[7] G. Lavoué, E. D. Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi, "Perceptually driven 3D distance metrics with application to watermarking," in *Proc. of the SPIE Electronic Imaging*, 2006, vol. 6312, pp. 63120L.1–63120L.12.

[8] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Process.*, vol. 13, no. 4, pp. 1–14, 2004.

[9] C. Zhang and T. Chen, "Efficient feature extraction for 2D/3D objects in mesh representation," in *Proc. of the IEEE Int. Conf. on Image Process.*, 2001, pp. 935–938.

[10] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "Robust and blind watermarking of polygonal meshes based on volume moments," Tech. Rep., LIRIS Laboratory - M2DisCo Team, 2009, available at http://liris.cnrs.fr/Documents/Liris-3713.pdf.

[11] G. Taubin, "Geometric signal processing on polygonal meshes," in *Proc. of the Eurographics State-of-the-art Reports*, 2000, pp. 81–96.

[12] P. Lindstrom and G. Turk, "Fast and memory efficient polygonal simplification," in *Proc. of the IEEE Visualization*, 1998, pp. 279–286.

[13] D. Zorin and P. Schröder, "Subdivision for modeling and animation," in *Proc. of the ACM Siggraph Course Notes*, 2000.

[14] J. W. Cho, R. Prost, and H. Y. Jung, "An oblivious watermarking for 3D polygonal meshes using distribution of vertex norms," *IEEE Trans. on Signal Process.*, vol. 55, no. 1, pp. 142–155, 2007.