A BENCHMARK FOR 3-D MESH WATERMARKING

Kai Wang[1]*, Guillaume Lavoué*[1]*, Florence Denis*[2]*, and Atilla Baskurt*[1]

[1,2] Université de Lyon, CNRS
[1] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[2] Université Lyon 1, LIRIS, UMR5205, F-69622, France
{kwang, glavoue, fdenis, abaskurt}@liris.cnrs.fr

ABSTRACT

This paper presents a benchmark for the evaluation of 3-D mesh watermarking methods. It comprises a data set, a software tool and two evaluation protocols. The data set contains several "standard" mesh models on which we suggest to test the watermarking algorithms. The software tool integrates both objective and perceptual measurements of the distortion induced by watermark embedding, and also the implementation of a variety of attacks on watermarked meshes. Besides, two different application-oriented evaluation protocols are proposed, which define the main steps to follow when conducting the evaluation experiments. Two state-of-the-art algorithms are tested and compared by using the proposed benchmarking framework.

Index Terms**—** Mesh watermarking, benchmarking, distortion, attack, robustness

1. INTRODUCTION

A 3-D mesh is a collection of polygonal facets targeting to constitute a piecewise linear approximation of the surface of a real 3-D object. It has three different combinatorial elements: vertices, edges and facets (typically triangles or quadrangles). The coordinates of the vertices constitute the *geometry* information of the mesh, while the edges and facets describe the adjacency relationship between the vertices and constitute the mesh's *connectivity* information.

3-D meshes are now more and more used in applications such as medical imaging, computer aided design and digital entertainment due to their algebraic simplicity and high usability. Unfortunately, like digital images and audio/video clips, mesh models can be easily duplicated and redistributed without any loss of quality by a pirate. This illegal behavior infringes the intellectual property of mesh owners and could also do harm to the whole underlying commercial chains. Actually, the generation of mesh models, either by scanning real 3-D objects or by using specific design software, is normally a time-consuming and expensive work. The robust watermarking technique appears as a good solution to the copyright protection problem of 3-D mesh models. This technique embeds 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 mesh watermarking methods have been proposed. These methods embed the watermark either directly in the spatial domain [1–3] or in a transformed spectral-like domain [4, 5] of the cover mesh. Readers could refer to [6] for a comprehensive survey on 3-D mesh watermarking.

This paper focuses on the evaluation of robust mesh watermarking schemes. Indeed, when a new robust scheme is proposed, we often want to compare it with some existing methods so as to fairly access 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. In this paper, we present a benchmarking software tool for the evaluation of robust mesh watermarking algorithms and introduce two application-oriented performance assessment protocols, with the objective to facilitate the experimental comparisons between different schemes.

Actually, it is almost impossible to assess the performance of a watermarking algorithm completely through theoretical analysis. Therefore, researchers often have to rely on a benchmarking system combined with a commonly used protocol to conduct an experimental evaluation. Several benchmarking tools and protocols have been proposed for image watermarking evaluation, such as Stirmark [7], Checkmark [8] and Optimark [9]. Contrarily, to the best of our knowledge, the benchmarking of 3-D mesh watermarks was only addressed by Bennour and Dugelay [10]. They propose to use some existing software packages to measure the objective distance between cover and watermarked models, and to exert attacks on watermarked meshes. The authors also propose a formula to calculate a final score as the robustness evaluation result and suggest a four-element structure to report the overall performance of a robust mesh watermarking scheme. Compared to their proposal, our contributions are threefold: (1) We provide a publicly available dataset collection of 3-D mesh models and a software tool for the purpose of mesh watermark evaluation[1]. The provided software comprises a large number of attacks, a perceptual distortion metric and the legacy implementation of several largely used objective distortion measurements. (2) Two protocols are defined for the capacity-distortion-robustness evaluation. In this way, researchers only need to provide some brief tables to report the performance of their watermarking schemes. The comparison then becomes easy and reliable since we all use the same models, distortion metrics and robustness evaluation methodology. (3) Two recent robust mesh watermarking algorithms are compared by using the proposed benchmarking software and protocols. 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 our mesh watermarking benchmark; Sections 3 and 4 present respectively the distortion metrics and the attacks integrated in our benchmarking software tool; we propose two different application-oriented evaluation protocols in

This work is in part supported by China Scholarship Council of the Chinese government and by the French National Research Agency (ANR) through Collaviz project (ANR-08-COSI-003-12).

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

Section 5; the evaluation results of two state-of-the-art algorithms, obtained by using the proposed benchmark, 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: *capacity*, *distortion*, *robustness* and *security*. The *capacity* is the number of bits of the hidden message conveyed by the watermark. The *distortion* measures the difference between the original cover content and its watermarked version. Note that this induced distortion can be measured either objectively or perceptually. The *robustness* indicates how resistant the watermarking scheme 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 mesh watermarking benchmark, we only considers the capacity, 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 [6] and until now the community has been interested in achieving robustness against connectivity attacks (*e.g.* surface simplification, subdivision 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.

Practically, in order to evaluate a robust mesh watermarking scheme by using the above metrics, we need a well-defined protocol that 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 the mesh watermark embedding procedure 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 *objectively* or *perceptually*. For the objective measurement, we propose to use the maximum root mean square error (MRMS). In general, the root mean square error (RMS) from one 3-D surface S to another 3-D 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. The calculation of MRMS has been implemented in some free software tools such as Metro [11] and MESH [12]. We included the implementation of Metro in our benchmarking software.

However, it is well known that the objective surface-to-surface distances do not correctly reflect the visual difference between two meshes. Thus, we need a perceptual metric to measure the visual distortion induced by the watermark embedding. For this purpose, we have considered the mesh structural distortion measure (MSDM) proposed by Lavoué *et al.* [13], and have integrated it in the benchmarking software. This metric follows the concept of structural similarity recently introduced by Wang *et al.* [14] for 2-D image quality assessment, and well reflects the perceptual distance between two 3-D 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:

$$L(p,q) = \frac{\|\mu_p - \mu_q\|}{\max(\mu_p,\mu_q)}, \tag{4}$$

$$C(p,q) = \frac{\|\sigma_p - \sigma_q\|}{\max(\sigma_p,\sigma_q)}, \tag{5}$$

$$S(p,q) = \frac{\|\sigma_p\sigma_q - \sigma_{pq}\|}{\sigma_p\sigma_q}, \tag{6}$$

with μ_p, σ_p and σ_{pq} respectively the mean, standard deviation and covariance of the curvature over the mesh local windows. The global MSDM measure between two meshes $\mathcal{M}$ and $\mathcal{M}^{'}$, is defined by a Minkowski sum of their n local window distances:

$$d_{MSDM}(\mathcal{M},\mathcal{M}^{'}) = \left(\frac{1}{n}\sum_{j=1}^{n} d_{LMSDM}(p_j,q_j)^3\right)^{\frac{1}{3}} \in [0,1). \tag{7}$$

Its value tends toward 1 (theoretical limit) when the measured objects are visually very different and is equal to 0 for identical ones. 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 [13].

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 attacks and present the corresponding implementations in our benchmarking software. The types and parameters of the attacks exerted on a given mesh can be adjusted through a configuration file.

4.1. File attack

This attack consists in reordering the vertices and/or the facets in the mesh file, and it 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 is kept unchanged. Our benchmarking software comprises 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):

$$x_i^{'} = x_i + a_i.\bar{d}, \tag{8}$$

where $\bar{d}$ denotes the average distance from vertices to mesh center, and a_i is the noise strength for x_i. The mesh center is calculated by using the analytic and continuous volume moments of the mesh [15], which is much more robust than simply calculating it as the average position of the mesh vertices. This robust mesh center calculation ensures a same level of induced distortion when the watermark embedding changes the mesh connectivity or when the noise addition is combined with a connectivity modification. 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 version of the original Stanford Bunny model that is shown in Fig. 1.(a).

Smoothing. Surface smoothing is a common operation used to remove the noises introduced during the mesh generation process through 3-D scanning. For the purpose of mesh watermark benchmarking, we choose to use a Laplacian smoothing [16] with different iteration numbers N_{itr} while fixing the deformation factor λ as 0.10. Figure 1.(c) shows a smoothed Bunny model.

Vertex coordinates quantization. This operation is largely used in lossy 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 eligible quantized levels. Figure 1.(d) illustrates a Bunny model whose vertex coordinates are quantized.

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 the one obtained by a 3-D scanning) often 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 model 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 [17], 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.(e) shows a simplified Bunny model.

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. The watermark robustness is tested against three typical subdivision schemes, always with one iteration: the simple midpoint scheme, the $\sqrt{3}$ scheme and the Loop scheme [18]. Note that the midpoint scheme adds vertices in the middle of the existing edges, and also edges within the existing facets. This subdivision scheme, which may be performed by a pirate as an attack, does not introduce any distortion to the test model; thus, ideally a robust mesh watermark should be invariant to it. Figure 1.(f) illustrates a subdivided Bunny model.

Cropping. In this attack, one part of the watermarked mesh is cut off and thus lost. This attack could happen 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.(g) shows a cropped Bunny model.

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 [7] propose to first fix the watermark capacity at about 70 bits and also to limit the induced distortion to be less 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 basically indicate the robustness metric (*e.g.* message error rate) 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 of establishing two different protocols is that different mesh-based applications have very different restrictions on the objective and 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 objective distortion is less important. On the contrary, for the meshes used in computer aided design and medical imaging, it is often required that the objective 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 objective distortion $d_{MRMS} \leq 0.08\%.l_{bbd}$, where l_{bbd} denotes the diagonal length of the mesh's bounding box.
2. Carry out the suggested attacks listed in Table 1 on the stego-mesh $\mathcal{M}^{'}$, by using the proposed benchmarking software.
3. Try to extract/detect the embedded watermark W from the obtained attacked stego-models and record the extraction/detection robustness evaluation results.
4. Repeat steps 1-3 for several times with different randomly selected watermark sequences and secret keys.
5. Repeat steps 1-4 for each test mesh from the standard dataset collection available at the benchmark website.

The selection of the two distortion thresholds for the perceptual quality oriented protocol ensures that the obtained stego-model

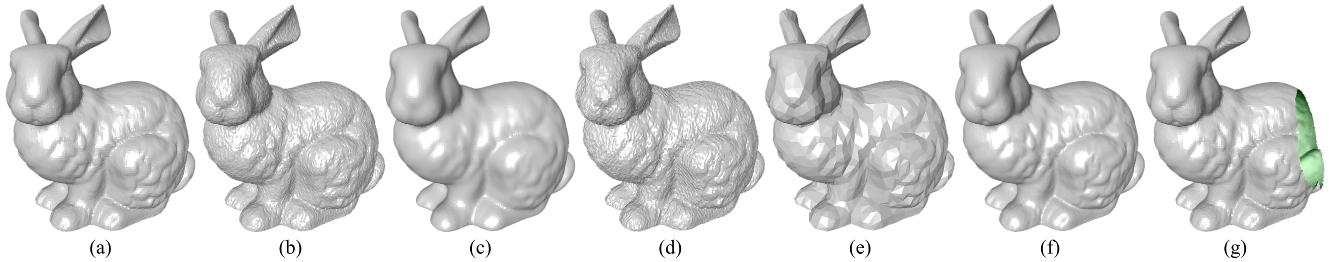

Fig. 1. The Stanford Bunny model and six attacked versions: (a) the original mesh having 34835 vertices and 104499 edges; (b) after noise addition ($A = 0.30\%$); (c) after Laplacian smoothing ($\lambda = 0.10, N_{itr} = 30$); (d) after vertex coordinates quantization ($R = 8$); (e) after simplification ($E_{sim} = 95\%$); (f) after subdivision (1 iteration, Loop scheme); (g) 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 ≥100K vertices.

Table 2. Baseline evaluation results of the two methods.

Methodology	Protocol A		Protocol B	
Method	Cho's	Wang's	Cho's	Wang's
WM capacity (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

is of very high visual quality and meanwhile prevents deforming too much the cover mesh. The geometric quality oriented protocol consists of the same steps except for that we have different constraints on the induced objective 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 can be 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 deformed object. We are prepared to adjust these four thresholds according to feedbacks from the research community. Finally, note that the two MSDM distance thresholds in the protocols are for the calculation where the radius parameter is equal to 0.005 [13].

Both detectable and readable watermarking schemes can be tested by using our protocols. For detectable schemes, it is suggested that for each test model we repeat the watermark embedding for at least 100 times by using different watermark sequences and keys. The receiver operating characteristics (ROC) curves under each kind of attacks are plotted as the evaluation results. 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.

As pointed out in [6], robust mesh watermarking is a challenging task due to many particular difficulties and the relevant research is still in its early stage. We have taken into account this point when proposing the evaluation protocols, which are actually much less stringent compared to the protocols for image watermarking evaluation. First, it is acceptable that a readable mesh watermarking scheme has a relatively low capacity. Indeed, the amount of capacity heavily depends on the application. Low-capacity schemes can also be very useful, for example in the application of copy control examination. We propose to set the capacity to one of the following values: 16 bits, 32 bits, 64 bits and ≥96 bits. However, when doing comparison between different schemes, we always have to ensure that they have a same capacity. Second, instead of message error rate, we adopt the bit error rate as the robustness evaluation metric for readable mesh watermarking algorithms. If the message error rate were used, the decoding process would only output 0 (failure) or 1 (success) and a multi-bit message would be considered successfully decoded only if all the bits were correctly retrieved. We think that this evaluation metric is too stringent considering the state of the art in mesh watermarking and that it is more appropriate to use the bit error rate as the metric.

Finally, concerning the dataset collection, we have selected several representative meshes (with different vertex numbers and different shape complexities) as the 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) and Rabbit (70658 vertices).

6. COMPARISON OF TWO RECENT ALGORITHMS

In order to test the utility of the proposed benchmarking software tool and protocols, we have used them to evaluate and compare two recent blind and robust readable mesh watermarking schemes: the method of Cho *et al.* [2] that is based on modification of the mean value of the vertex norm histogram and the method of Wang *et al.* [3] that is based on modification of the mesh local volume moments. Table 2 presents the baseline evaluation results of the two methods under the two protocols on the Venus model. Some of the robustness evaluation results (always on Venus model) are presented in Table 3. In both tables, "Protocol A" represents the perceptual quality oriented protocol and "Protocol B" stands for the geometric quality oriented protocol. All the results are the averages of 5 trials with randomly selected watermark sequences and keys. From these results, we can conclude that, for this model, 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 induced objective distortion. However, in both kinds of applications, if a strong robustness against connectivity attacks is required, then the method of Wang *et al.* seems a better choice. In all, the advantage of the method of Cho *et al.* is that the watermark can resist attacks that introduce much higher objective distortions than its embedding, and the main strengths of the method of Wang

Table 3. Robustness comparison between the two methods.

Methodology ⇒	Protocol A		Protocol B	
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 $R = 10$	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
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 $E_{sim} = 50\%$	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

et al. are its strong robustness against connectivity attacks and its high watermark imperceptibility.

7. CONCLUSION

We proposed a benchmark for the evaluation of 3-D mesh watermarking schemes. MRMS is used to measure the objective distortion induced by the watermark embedding, while the perceptual distortion is evaluated by MSDM. A software tool including these two distortion metrics, as well as a large number of attacks, is implemented and made publicly available. Two mesh watermarking evaluation protocols are established: the perceptual quality oriented protocol is designed for the applications which require a high visual quality of the watermarked model and the geometric quality oriented protocol is to be used in the applications which have strict restriction on the induced objective distortion. Two recent 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 (http://liris.cnrs.fr/meshbenchmark/), hence we expect that the scientific community will contribute by adding new test meshes, new attacks and providing other protocols.

8. ACKNOWLEDGMENTS

We would like to thank Pr. 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. We are also very grateful to Dr. P. Cignoni for allowing us to integrate the Metro tool in our benchmarking software.

9. REFERENCES

[1] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Trans. on Vis. and Comput. Graphics*, vol. 11, no. 5, pp. 596–607, 2005.

[2] 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.

[3] 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.

[4] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Comput. Graphics Forum*, vol. 21, no. 3, pp. 373–382, 2002.

[5] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "Hierarchical watermarking of semiregular meshes based on wavelet transform," *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 4, pp. 620–634, 2008.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measuring errors between surfaces using the Hausdorff distance," in *Proc. of the IEEE Int. Conf. on Multimedia & Expo*, 2002, pp. 705 – 708.

[13] 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.

[14] 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.

[15] 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.

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

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

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