Toward a 3D watermarking benchmark

Jihane Bennour, Jean-Luc Dugelay
Institut Eurécom
2229 route des crêtes, BP 193
06904 Sophia Antipolis, France
Email: {bennour, dugelay}@eurecom.fr

***Abstract*— In the last few years, a large number of $3D$ watermarking schemes have been proposed. We describe in this paper a possible benchmark to evaluate $3D$ watermarking algorithms. A list of objects and basic reproducible attacks against which $3D$ watermarking system could be evaluated are proposed as well as a way to compute a final score.**

I. Introduction

$3D$ Watermarking is a hot topic in the watermarking community. Similar to image watermarking, $3D$ watermarking aims at hiding in an invisible way information inside a $3D$ object. This information can then be recovered at any time, even if $3D$ object was altered by one or more non destructive attacks, either malicious or not. $3D$ watermarking can be useful in several applications, security-related ones being the most prominent. For example, users would like to check if the use of a given object is legal or not, to access additional information concerning the object (e.g for authentication or indexing), the owner (copyright), or even the buyer (e.g for traitor tracing).

The fact that the creation of $3D$ models is, in many cases, a labor-intensive and costly procedure made the protection of such models an urgent necessity and attracted the interest of many engineers and researches toward $3D$ watermarking. As a result, numerous watermarking algorithms for $3D$ objects have been proposed in the literature. The watermarking community needs therefore some advanced protocols in order to compare performances of diverse proposed $3D$ watermarking technologies.

By analogy with still image watermarking where some benchmarking tools such as Stirmark [17], Checkmark [3], Optimark [15] and Certimark [2] have been created to standardize watermarking system evaluating process. We propose in this paper a way to evaluate the performances of $3D$ watermarking techniques. First we recall in section II a few prerequisite notions of watermarking $3D$ objects, then we present in section III our protocol to evaluate $3D$ watermarking techniques and we conclude in section IV with some open discussions.

II. Basic review on $3D$ watermarking

In this section, we recall some basic watermarking notions, dedicated to the special case of 3D objects.

By analogy with still image watermarking which was historically the first type of digital documents investigated for watermarking, 3D watermarking techniques can be separated in two major categories with respect to the information conveyed by the watermark:

- *Zero bit.* Watermarking techniques in this category can only verify whether the data is watermarked or not.
- *Multiple bit.* The capacity of a multiple bit watermarking algorithm is the amount of information, i.e., the length of the message, that can be hidden in the watermarked $3D$ object. It should be noted that, in most cases, data payload depends on the size of the host data. The more the available host samples, the more bits can be hidden. Thus, capacity is often given in terms of message bits per sample of the host $3D$ object. In such system the $3D$ object under investigation is first tested to verify whether it hosts a watermark or not. If the $3D$ object is indeed watermarked, the embedded message is decoded.

With respect to the method used for watermark detection, techniques can be classified according to the following categories:

- *Blind.* The only information required to extract the watermark from the $3D$ object under investigation is the watermark key.
- *Non-blind algorithms.* To extract the watermark, one should have not only the $3D$ object that is being checked, but also the original $3D$ object from which it is assumed to be derived. Obviously, the requirement that the original 3D object is available during watermark extraction implies serious limitations for the applicability of such algorithms in a number of scenarios. For example non-blind techniques cannot be used for the automatic search over Internet to detect illegal copies of one's 3D objects.

In a similar way, watermarking techniques can be distinguished into two major categories with respect to the way embedding is performed.

- *Blind embedding techniques.* Such techniques consider the host $3D$ object as noise or interference. In most cases, these methods utilize knowledge of the host signal statistics.
- *Informed coding/embedding techniques.* These techniques exploit the fact that during embedding, not only the statistics of the host $3D$ object but also the object itself is known and try to utilize this fact in order to improve watermark detection performance (e.g. through interference cancellation).

3D watermarking algorithms can also be classified according to the technique used to embed the watermark.

- *Data file organization*. These algorithms encode information by modifying the organization of the data in the computer file associated with the $3D$ object. To achieve this, authors typically propose to modify in the computer file, the order of the triangles within a list of triangles or the order of the triplet of vertices forming a given triangle. Readers can refer to the publications [7][14][12][4] for more details about this category of techniques.
- *Topological data*. These algorithms, that operate on mesh data, use the topology of the $3D$ object, i.e., the connectivity of the mesh to embed data. The geometry of the mesh, i.e., the positions of the vertices, is not modified [19][13].
- *Geometrical data*. These algorithms are based on slight modifications performed on the geometric data of the $3D$ object. Typically authors propose to modify the $3D$ coordinates of some points. Interested readers are invited to refer to the following papers [13][1][6][16][8].

III. Our evaluation protocol

After a $3D$ watermarking technique has been designed and implemented, it is important to quantitatively evaluate its performance. In this section, we propose a protocol to evaluate $3D$ watermarking techniques.

$3D$ watermarking algorithms should satisfy the trade-off between visibility and robustness. Our evaluation schema focus on these two most important attributes which conflict each other.

A. Visibility

The term visibility refers to the visual degradation of a $3D$ object due to the embedding of the watermark. Naturally, a watermark should be as invisible as possible and thus, the watermarking process should not introduce suspicious perceptible artifacts. In some other words, a human observer should not be able to detect if some digital data has been watermarked or not.

Some techniques have been introduced to measure objectively whether a distortion due to embedding is perceptible or not.

1) Compute the signal to noise ratio using the following formula:
$SNR =$

$$\frac{\sum_{i=1}^{N} x_i^2 + y_i^2 + z_i^2}{\sum_{i=1}^{N} (x_i - x_i^W)^2 + (y_i - y_i^W)^2 + (z_i - z_i^W)^2} \quad (1)$$

where N is the number of vertices of the $3D$ object, x_i, y_i and z_i are the coordinates of the vertex v_i before embedding and x_i^W, y_i^W and z_i^W are the coordinates of the same vertex after the embedding of the watermark.

2) Compute the Hausdorff distance (often used by default in $3D$ watermarking).

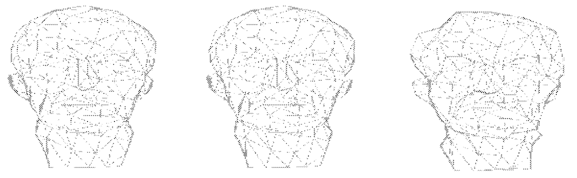

Fig. 1. Random noise applied to the geometry of a $3D$ mesh with different magnitude

$$d_H(X,Y) = max$$

$$\{sup_{x \in X}\ inf_{y \in Y}\ d(x,y)\ ,\ sup_{y \in Y}\ inf_{x \in X}\ d(x,y)\} \quad (2)$$

MESH [9] (Measuring Error between Surfaces using the Hausdorff distance) is a tool that measures distortion between two discrete surfaces (triangular meshes). It uses the Hausdorff distance to compute a maximum, mean and root-mean-square errors between two given surfaces. MESH is an open-source (GPL) software.

3) Evaluate 3D watermarking perceptual quality. Gelasca et al [5] have published the first paper for this particular problem. This technique relies on the objective observation scores given by a pool of viewers.

B. Robustness

Watermark robustness is the ability to recover the watermark even if the watermarked $3D$ object has been manipulated. In the case of $2D$ images, watermark robustness can be easily evaluated thanks to the existence of a standardized benchmark tests. To the best of our knowledge, no similar standards or objective techniques have been proposed for $3D$ mesh watermarking. We propose here some tests to better evaluate the robustness of $3D$ watermarking techniques. Those tests are designed to create various distortions to the watermarked cover under test, so that it is possible to measure watermark detection rate under those conditions and then to judge the performance of the watermarking algorithm.

1) Noise on geometry: Addition of noise consists in adding random noise to vertex coordinates (see Fig. 1). Such a noise could be introduced either maliciously to weaken the watermark or during typical $3D$ object manipulations like lossy compression or format conversion. We propose to use the open-source software available at [18] to noisify $3D$ vertex coordinates using various magnitudes.

2) Mesh smoothing: Meshes obtained from real world objects are often noisy. Mesh smoothing relocates mesh vertices to improve the mesh quality while keeping the mesh topology unchanged (see Fig. 2). Several techniques can be used for mesh smoothing, we can cite the umbrella-operator smoothing and the Taubin's lambda-mu smoothing. A software available at [18] can be used for $3D$ mesh smoothing.

3) Global transformations: An affine transform like translation, rotation, uniform or non-uniform scaling, or even a

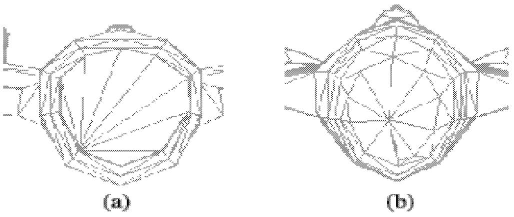

Fig. 2. Impact of the umbrella-operator smoothing in a 3D mesh. (a) before smoothing and (b) after smoothing.

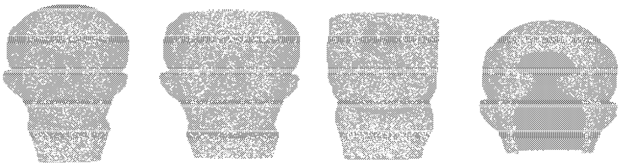

Fig. 3. Cropping of geometry.

projective transform or another non-affine global transform can be performed on a protected object. Some algorithms rely on the precise position and orientation of an object to extract the watermark. In this case, a global transformation applied, maliciously or not, to the object can prevent watermark detection unless a way is found to recover the object's original reference frame used for watermarking. For global transformations of a $3D$ mesh we propose to use the software available at [18].

4) Cropping: Similar to images, it is possible to remove part of an object's geometry (see Fig. 3). In some cases this would yield to a meaningless object, but in other cases the remaining part could still be of some value, whereas the watermark may be destroyed in the process. In fact, cropping would often destroy the structure of the watermark embedding space (e.g. the ordering of vertices), unless the watermarking algorithm is designed to be robust to cropping by hiding the watermark several times in different parts of the object. We propose to use the open-source at [18] to simulate cropping attacks.

5) Edge and face flip: Edge flip consists on optimally re-triangulate a mesh in order to minimize the maximum angle in a given triangle (v_1, v_2, v_3) (see Fig. 4). Whereas face flip which is one possibility of vertex-reordering affects the data organization within the file. It consists in flipping the vertices order in each face, the shape of the $3D$ object itself remains the same. We propose to use the software available at [18] for these manipulations.

6) Mesh simplification: The goal of mesh simplification is to speed up manipulations and rendering of $3D$ objects. It consists in displaying the 3D polygonal mesh with fewer triangles while preserving the shape. It is a challenging attack for $3D$ watermarking community. For this aim, one can use Michael Garland's QSlim mesh simplification software available at [10].

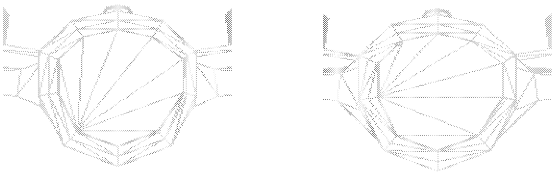

Fig. 4. Edge flip.

C. Evaluation

In order to evaluate $3D$ watermarking techniques and have a fair comparison between techniques, we propose the following protocol.

$3D$ watermarking techniques have to be tested on various $3D$ objects using different keys. For *robustness performance*, the $3D$ watermarking algorithm is tested under the real-world attacks presented in section III-B. For *zero bit* techniques, resistance to these attacks is quantitatively evaluated by Equal Error Rate which indicates that the proportion of false acceptances (i.e. the probability of detecting a watermark on a non watermarked model or a watermarked model with another key) is equal to the proportion of false rejections (i.e. the probability of not detecting a watermark in a watermarked model using the correct key). The lower the equal error rate value, the higher the performance of the system. In case of *multiple bit* techniques robustness can be evaluated by Message Error Rate (MER).

The *visual quality* is evaluated using one of the distortion metrics presented in section III-A.

Parameter settings. In this paragraph we propose a set of parameters to compute the score of a $3D$ watermarking algorithm. $3D$ watermarking algorithms have to be tested on the 5 models listed in table I and available at [11]. First, we evaluate the distortion using the Hausdorff metric. Then, for *zero bit* technologies, for each attack and each level presented in table II, detection must be performed using 1000 correct and 1000 wrong keys. We have a binary $Extraction_{result}$, 1 if the detection is performed correctly, that is to say the computed EER is less than 10^{-4} and 0 otherwise. For *multiple bit* technologies, $Extraction_{result}$ is equal to 1 if the message is perfectly extracted and 0 otherwise. The final $Robustness_{score}$ is a scale between 0 and 20 computed as follows:

$$\frac{1}{6}\sum_{Object}\sum_{Attack}\sum_{Level} Extraction_{result}. \tag{3}$$

Just to note that for this first version of the benchmark we consider that attacks have the same importance. We did not yet take into account to perform combined attacks on the $3D$ model that could be still much more challenging. For the noise addition attack, the percentage represents the noise amplitude as a fraction of the largest dimension of the object. For mesh smoothing, the parameters 2, 10, 30 represent how many iterations are applied using the umbrella-operator smoothing.

Rotation is done according to each axis. Face flip attack changes the order of vertices within each face.

It is worth mentioning that in addition to visual quality and robustness, $3D$ watermarking algorithm include some other properties that could be evaluated as capacity (the amount of information being embedding) and the method used for detection (blind or non-blind). The *final score* of the proposed $3D$ watermarking algorithm has therefore the following structure:

$$Capacity - Distortion - Robustness_{score} - Blind/nonblind. \tag{4}$$

The distortion is the average distortion computed on the 5 protected objects.

An interesting specific score could consist in setting a default threshold for the visual quality of the watermarked object. That is to say we impose that the distortion due to watermarking on each object according to the Root Mean Square computed with the MESH tool is less than 0.06. The *final score* in this case will have the following structure:

$$Capacity - 0.06 - Robustness_{score} - Blind/nonblind. \tag{5}$$

3D model	Vertices	Faces
Bunny	16760	33518
Isis	46969	93934
Dragon	437645	871414
Venus	134359	268714
Horse	48485	96966

TABLE I

3D MODELS.

Attacks	Low	Medium	High
Noise imposition [a]	0.2%	0.5%	0.7%
Mesh smoothing [b]	2	10	30
Edge flip [c]	-	-	-
Face flip [d]	-	-	-
Rotation [e]	2^o	10^o	180^o
Scaling [f]	$\frac{2}{3}, \frac{3}{2}$	$\frac{1}{2}, 2$	$\frac{1}{5}, 5$
Cropping [g]	$\frac{1}{8}$	$\frac{1}{3}$	$\frac{1}{2}$
Mesh simplification [h]	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{8}$

[a] cmd. infile -noisify 0.02 outfile

[b] cmd. infile -usmooth 2 outfile

[c] cmd. infile -edgeflip outfile

[d] cmd. infile -faceflip 2 outfile

[e] cmd. infile -rot 2 1 1 1 outfile

[f] cmd. infile -scale 2/3 outfile

[g] cmd. infile -clip bbox (file with 6 numbers) outfile

[h] cmd. Qslim -t 2/3*number of faces infile -o outfile

TABLE II

ROBUSTNESS SCORE. SIMILAR TO STIRMARK, DIFFERENT LEVELS OF ATTACKS HAVE BEEN CHOSEN AS EASY (LOW), INTERMEDIATE (MEDIUM) ANS DIFFICULT TO RESIST (HIGH).

In short, to evaluate a $3D$ watermarking algorithm users have to download the software available at [9] to measure objectively the distortion due to embedding. Then, they have to compute the $Robustness_{score}$ (Eq. 3) of their technique according to the 8 attacks available at [18] and [10] and using the parameters presented in table II. Finally they have to mention the capacity of their algorithm and the method used for detection (blind/non blind). All the tests have to be carried out on the 5 models listed in table I and available at [11].

IV. CONCLUSION

In this paper, we have addressed the important and often neglected issue of evaluating $3D$ watermarking techniques.

Our evaluation protocol focus on the two most important attributes of robustness and visibility. We have presented a list of real-world attacks against which $3D$ watermarking system could be judged. A link for an-open source software is provided for all the presented attacks. Some techniques have also been presented to objectively measure whether a distortion due to embedding is perceptible or not. Just to note that to evaluate distortion we have adopted the Hausdorff distance as an open-source software is already available but some other metrics could be used as well.

REFERENCES

[1] A. G. Bors. Blind watermarking of 3D shapes using localized constraints. *IEEE 3DPVT*, pages 242–249, Sept 2004.

[2] Certimark. www.certimark.org/.

[3] Checkmark. www.watermarking.unige.ch/checkmark/.

[4] C. Fornaro and A. Sanna. Public key watermarking for authentication of CSG models. *Computer Aided Design*, 32(12):727–735, 2000.

[5] D. Gelasca, T. Ebrahimi, and M. Corsini nd M. Barni. Objective Evaluation of the Perceptual Quality of 3D Watermarking. In *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2005.

[6] S. Hanai, H. Date, and T. Kishinami. Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In *6th IFIP 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6)*, pages 296–307, Tokyo, Japan, December 1998.

[7] S. Ichikawa, H. Chiyama, and K. Akabane. Redundancy in 3D polygon models and its application to digital signature. *Journal of WSCG*, 10(1):225–232, 2002.

[8] J. J. Lee, N. I. Cho, and J. W. Kim. Watermarking for 3D NURBS graphic data. In *IEEE International Workshop on MultiMedia Signal Processing*, December 2002.

[9] MESH. www.mesh.berlios.de/.

[10] Qslim mesh simplification software. http://graphics.cs.uiuc.edu/ garland/software/qslim.html.

[11] 3D models. www.mathinfo.univ-reims.fr/image/pagebuilder.php?dir= initiation&show=dataply&menu=base.

[12] R. Ohbuchi and H. Masuda. Managing cad data as a multimedia data type using digital watermarking.

[13] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models. In *ACM Multimedia*, pages 261–272, Seattle, Washington, November 1997.

[14] R. Ohbuchi, H. Masuda, and M. Aono. A shape-preserving data embedding algorithm for NURBS curves and surfaces. In *Computer Graphics International*, pages 170–177, June 1999.

[15] Optimark. www.poseidon.csd.auth.gr/optimark/.

[16] H. S. Song, N. I. Cho, and J. W. Kim. Robust watermarking of 3D mesh models. In *IEEE International Workshop on MultiMedia Signal Processing*, December 2002.

[17] Stirmark. www.watermarkingworld.org/.

[18] trimesh. www.graphics.stanford.edu/software/trimesh/.

[19] I. Weiss and M. Ray. Model-based recognition of 3d objects from single images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(2):116–128, 2001.