Robust and Blind Watermarking of Polygonal Meshes Based on Volume Moments

Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt

Abstract

This paper describes a robust and blind watermarking algorithm for polygonal meshes, dedicated to copyright protection. The watermark primitives are intrinsic 3D shape descriptors: the analytic and continuous geometric volume moments. The mesh is first normalized by using its global moments, and decomposed into patches by discretizing its cylindric domain. Then, one bit is inserted in each candidate patch by quantizing its local zero-order moment, through a modified scalar Costa scheme. The patch is then deformed by an iterative process, so as to reach the target quantized moment value; a smooth deformation mask is used to avoid introducing visible distortion. A global moment compensation post-processing is carried out after bit insertion so as to recover the normalized mesh pose; thus, the causality problem is resolved. The watermarking security is ensured by the key-dependent scalar Costa quantization. The blind watermark extraction simply consists of mesh normalization, patch decomposition and bit extraction. Experimental results and comparisons with the state-of-the-art have demonstrated the superiority of the proposed approach in terms of robustness, security and imperceptibility. Moreover, to the authors' knowledge, it is the first attempt in the literature to tackle the robustness against 3D representation conversions (e.g. discretization of the mesh into voxels).

Index Terms

Polygonal mesh, watermarking, robustness, blindness, volume moment, imperceptibility, causality problem, security.

I. Introduction

RECENT advances in 3D acquisition technologies, 3D graphics rendering and geometric modeling have boosted the creation of 3D model archives in many applications including cultural heritage, medical imaging, virtual reality, video games, computer-aided design and so forth. Moreover, with the development of 3D graphic hardware, high capacity mobile devices and with the technological advances in telecommunication, 3D models are now commonly manipulated, visualized and transmitted over the Internet or intranets. With the increasing diffusion of these 3D models over networks along with their increasing complexity (i.e. added-value), there now exists a critical demand for protecting their intellectual property against unauthorized duplication and distribution. Digital watermarking [1], [2] is considered as an efficient solution for the copyright protection of this 3D content, mostly represented by polygonal meshes.

The basic idea of digital watermarking is to hide a piece of secret information within the functional part of a cover content, which could be an image, an audio or video clip, a software package or a 3D model. The main applications of this promising technique are the intellectual property protection and the authentication of various multimedia contents. The watermarks used for intellectual property protection are often called *robust* watermarks. Such a watermark is supposed to be as resistant as possible against both routine operations and malicious attacks on the watermarked content, while keeping itself imperceptible. A watermarked multimedia content can still be protected after the transmission phase and the legal access, since the inserted watermark always travels along with it. According to whether the original non-watermarked content is required or not at the watermark extraction, watermarking techniques are classified into two groups: *non-blind* and *blind*. The blind methods are preferred in real-world applications since in many scenarios the original non-watermarked content is difficult to or even cannot

K. Wang, G. Lavoué, and A. Baskurt are with Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France. (e-mail: kwang@liris.cnrs.fr, glavoue@liris.cnrs.fr, abaskurt@liris.cnrs.fr)

F. Denis is with Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France. (e-mail: fdenis@liris.cnrs.fr)

be available at extraction. One example is the multimedia copy control application at the client side: it is not suitable to make the original version available at the control device that is on the hand of a possibly non-reliable client.

This paper focuses on robust and blind watermarking of polygonal meshes; the design of such algorithms is difficult mainly because of the mesh irregular sampling nature and the existence of many intractable attacks [3]. Besides the *geometry attacks* that only modify the vertex positions (e.g. similarity transformation, noise addition, smoothing and vertex coordinate quantization), the *connectivity attacks* (e.g. simplification and remeshing) can completely change the geometry and connectivity information of the watermarked mesh while conserving its global 3D shape. The most destructive attack may be the *object representation conversion* (e.g. from mesh to voxels); the mesh itself disappears after such a conversion. These attacks will be potentially conducted by pirates who attempt to remove the watermark (i.e. the copyright proof) from the model; meanwhile, these attacks also try to conserve the global 3D shape (i.e. the visual appearance) of the model, since this constitutes its main added-value. Indeed, a too much distorted object does not present any interest for the pirate. Following this idea, we believe that a valuable robust watermark has to be linked to the *3D shape* that is behind the mesh, and not to the mesh itself. Hence, we have chosen an intrinsic 3D shape descriptor as watermarking primitive: the analytic and continuous geometric volume moment. This descriptor is calculated using combinatorial elements of the mesh but depends only on its 3D analytic shape and thus should be robust to geometry, connectivity and representation attacks providing that they are not too destructive for the object. In our method, a readable multi-bit blind watermark is inserted by slightly modifying these geometric moments through an informed quantization data hiding scheme that is widely used in image, audio and video watermarking.

Another critical issue for 3D mesh blind watermarking is the *causality problem*, which means that the posteriorly inserted watermark bits disturb the correctness and/or the synchronization of the previously inserted ones. For instance, in [4], the author establishes an order for the watermarking candidate vertices according to a geometric criterion, and then modifies another correlated geometric metric to insert watermark bits. The established order may be altered after the bit insertion; that is the reason why the author introduces a post-processing step in order to recover the original vertex order. In our algorithm, after watermark insertion, we introduce a geometric *compensation* process so as to resolve this causality problem, by restoring the initial mesh features. Two other points have also to be taken into account. The first is the watermark *imperceptibility*. It has been demonstrated that insertion in the mesh low-frequency components can be both more robust and more imperceptible [5]. We have followed this principle when devising our method. The second is the watermarking *security*. In the early stage of watermarking research, a good security level meant preventing non-authorized extraction and optimal watermark removal. Recent results [6] reveal that we have also to keep the information leakage of the secret parameters (usually determined by a secret key) of the watermarking system as low as possible. However, in 3D mesh watermarking, the security has often been ignored until now; while in our scheme, the security is explicitly taken into account.

Hence, we present a new robust and blind mesh watermarking algorithm based on volume moments. First, the mesh is normalized to a canonical and robust pose according to its global volume moments. Then, the mesh is decomposed into patches and a multi-bit readable watermark is inserted by quantizing the zero-order volume moments of some selected candidate patches. Experimental results and comparisons with the state-of-the-art have demonstrated the superiority of our approach in terms of robustness (against geometry attacks, connectivity attacks and object representation conversions), security and imperceptibility of the watermark. The remainder of this paper is organized as follows: section II introduces the related work and the geometric volume moments; section III provides an overview of the proposed method; sections IV and V detail the watermark insertion and extraction procedures; section VI presents some experimental results; finally section VII concludes the paper and gives some future working directions.

II. Related Work and Background

A. Robust and Blind Mesh Watermarking

A robust and blind watermark does not need the original non-watermarked cover content for its extraction, and has to resist the attacks that cause distortions under a certain threshold beyond which the watermarked content is greatly degraded. Relatively few robust and blind watermarking schemes have been proposed for polygonal meshes mainly due to the particular difficulties mentioned above (e.g. irregular sampling and connectivity attacks).

The blindness has been achieved in several *spatial-domain-based* mesh watermarking algorithms [4], [7], [8]; however, these schemes are not robust, especially against connectivity attacks because their geometric watermarking

$$m_{000}^{(f_i)} = \frac{1}{6}\left|x_{i1}y_{i2}z_{i3} - x_{i1}y_{i3}z_{i2} - y_{i1}x_{i2}z_{i3} + y_{i1}x_{i3}z_{i2} + z_{i1}x_{i2}y_{i3} - z_{i1}x_{i3}y_{i2}\right| \quad (2)$$

$$m_{100}^{(f_i)} = \frac{1}{4}(x_{i1} + x_{i2} + x_{i3}).m_{000}^{(f_i)} \quad (3)$$

$$m_{200}^{(f_i)} = \frac{1}{10}(x_{i1}^2 + x_{i2}^2 + x_{i3}^2 + x_{i1}x_{i2} + x_{i1}x_{i3} + x_{i2}x_{i3}).m_{000}^{(f_i)} \quad (4)$$

$$m_{110}^{(f_i)} = \frac{1}{10}(x_{i1}y_{i1} + x_{i2}y_{i2} + x_{i3}y_{i3} + \frac{x_{i1}y_{i2} + x_{i1}y_{i3} + x_{i2}y_{i1} + x_{i2}y_{i3} + x_{i3}y_{i1} + x_{i3}y_{i2}}{2}).m_{000}^{(f_i)} \quad (5)$$

primitives disappear after such attacks. On the contrary, some *transform-domain-based* algorithms [9]–[12] are robust but non-blind. The used transformation tools are sensitive to connectivity changes; hence, a resampling pre-processing step is needed at extraction. This step ensures constructing the same connectivity as the cover mesh but inevitably makes the scheme non-blind. There exist several blind algorithms in a transform domain [13]–[16]; however, they are not robust enough against connectivity attacks.

Several blind and robust algorithms have been nevertheless proposed. In order to achieve robustness to connectivity attacks, they consider mesh shape descriptors as watermarking primitives: the average of the facet normals in a patch [17], the histogram of the vertex coordinate prediction errors [18], and the vertex norm histogram [19]. These methods are either blind [18], [19] or semi-blind [17] and demonstrate good robustness due to the intrinsic stability of the shape descriptor primitives. The method of Cho et al. [19] may have been the most robust blind algorithm in the literature. However, it seems there exist two problems: first, this method has a low security level because the modified vertex norm histogram is exposed to everyone and a non-authorized extraction or an optimal removal can be easily carried out; second, the mesh center is calculated simply as the average of all its vertices, which is not very stable under non-uniform simplification and resampling.

B. Geometric Volume Moments

The geometric volume moments of a closed 3D surface are defined as:

$$m_{pqr} = \int\int\int x^p y^q z^r \rho\,(x,y,z)\,dxdydz \quad (1)$$

where p, q, r are the orders, and $\rho\,(x,y,z)$ is the volume indicator function (it is equal to 1 if (x,y,z) is inside the closed surface; otherwise it is equal to 0). For an *orientable* mesh, Zhang and Chen [20] and Tuzikov et al. [21] derived independently the explicit expression for the above volume integration. The basic idea is to calculate it as a sum of signed integrations over elementary shapes. For a 3D triangular mesh, the elementary shape is the tetrahedron constituted of a triangle facet f_i and the coordinate system origin $\mathcal{O}$. The contribution sign for each tetrahedron is determined by the orientation of f_i and the relative position between f_i and $\mathcal{O}$. Note that if the facets are correctly oriented, the moment m_{000} is the volume of the closed surface. Some of the low-order elementary moment integration expressions $m_{pqr}^{(f_i)}$ are listed as Eqs. 2 to 5, where $f_i = \{v_{i1}, v_{i2}, v_{i3}\} = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}), (x_{i3}, y_{i3}, z_{i3})\}$. With the above calculation, geometric volume moments can be easily generalized to non-closed orientable surfaces (e.g. a mesh patch). The calculation consists in first adding fictional facets by connecting the boundary vertices and the origin, and then calculating the moments of the obtained closed surface. These geometric moments are very robust features and have been used for mesh self-registration and retrieval [20]. In this paper, we use the global volume moments for mesh normalization and the local volume moments as watermarking primitives. Actually, invariant and orthogonal moments have already been used for robust image watermarking in [22]–[24].

III. Overview of the Proposed Method

Our method is based on the assumption that a good watermarking primitive has to be intrinsically linked to the 3D shape that is behind the mesh. The analytic and continuous volume moments seem good candidates. We wish to consider them as primitives to insert a *multi-bit readable* watermark (in contrast to a *detectable* watermark). Two difficulties immediately arise: first, the moments of different orders are correlated so it becomes very complicated

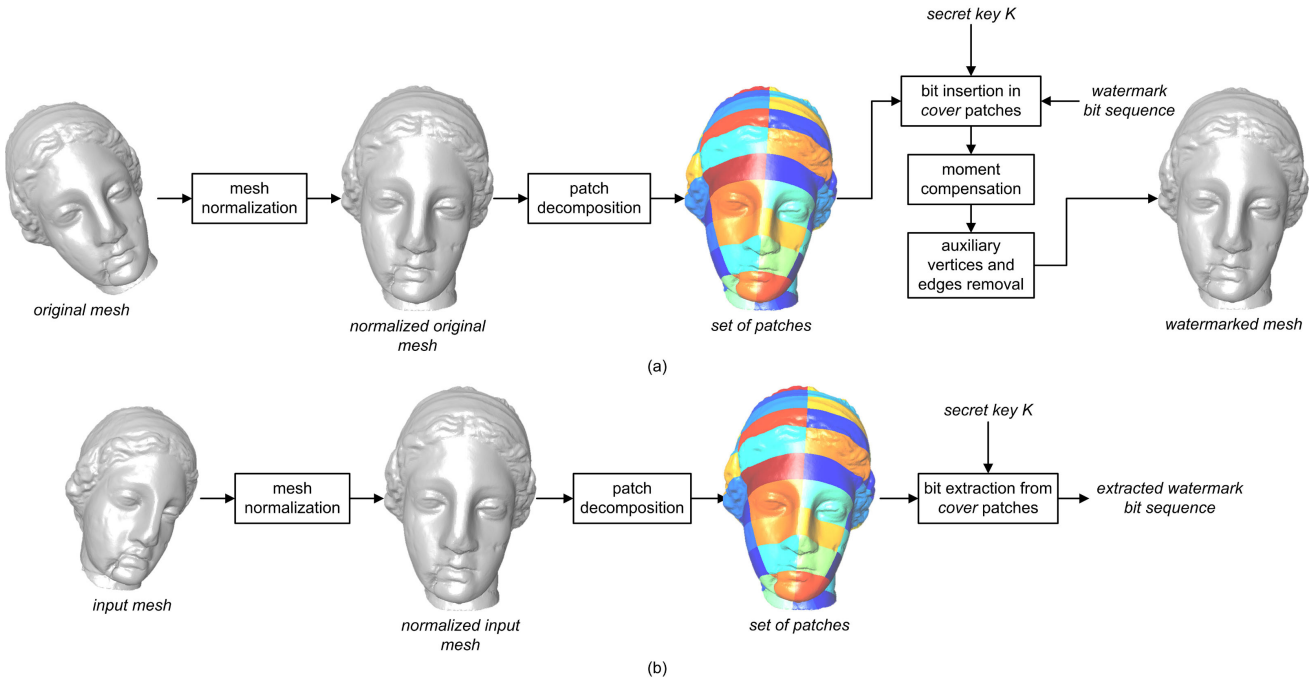

Fig. 1. Block diagrams of (a) the watermark insertion and (b) the watermark extraction procedures.

to modify different moments of a same mesh simultaneously and independently (in order to insert multiple bits); second, the above moment transformation is not reversible so we cannot modify the volume moment to a certain target value in an easy and straightforward way. The first point forces us to decompose the mesh into patches and insert one bit in each patch. For the second point, we introduce an iterative deformation algorithm for the patches, in order to reach their target moment values.

Figure 1.(a) illustrates the bloc diagram of our watermark insertion procedure. The cover mesh is first normalized by using its global volume moments. Then, the mesh is transformed from Cartesian coordinate system (x, y, z) to cylindric coordinate system (h, r, θ). The mesh is then decomposed into patches by discretizing the obtained h and θ domains. For several selected patches (*cover* patches), we calculate their zero-order moments and quantize them so as to embed one bit per patch. Note that in order to ensure a precise patch moment calculation, we insert some auxiliary vertices and edges on the patch borders; they can be easily removed after the watermark insertion. The moment modification is carried out through iterative patch deformation. Special care is taken in order to keep the deformation as imperceptible as possible. The third difficulty, namely the causality problem, occurs at this point, because after the deformation of the cover patches, the mesh global volume moments are probably altered so that we cannot achieve the same normalized mesh pose at extraction in a blind way. A moment *compensation* post-processing step is introduced to resolve this problem.

Figure 1.(b) illustrates the watermark extraction procedure. It does not require the original non-watermarked mesh nor any other supplementary information, except a secret key for reason of security. The extraction consists of three steps: mesh normalization, patch decomposition and watermark bits extraction from the cover patches. Following sections will present more details about the watermark insertion and extraction procedures.

IV. Watermark Insertion

A. Mesh Normalization

Mesh normalization is used as a preprocessing step by both watermark insertion and extraction, and consists of the following three sequential operations:

1) translation of the mesh so that its center coincides with the origin of the objective Cartesian coordinate system;
2) uniform scaling of the mesh so that it is bounded within a unit sphere;
3) rotation of the mesh so that its three principal axes coincide with the axes of the coordinate system.

TABLE I
ROBUSTNESS EVALUATION OF THE DIFFERENT NORMALIZATION SCHEMES ON THE VENUS HEAD MESH

Attack	Discrete moments		Surface moments		Volume moments	
	$\Delta \|\|C\|\|$	$\max\{\Delta PA\}$	$\Delta \|\|C\|\|$	$\max\{\Delta PA\}$	$\Delta \|\|C\|\|$	$\max\{\Delta PA\}$
0.50% noise	1.12×10^{-6}	$0.003°$	6.36×10^{-4}	$0.23°$	3.74×10^{-5}	$0.01°$
7-bit quantization	1.57×10^{-5}	$0.01°$	2.98×10^{-3}	$1.07°$	2.70×10^{-5}	$0.05°$
90% simplification	3.29×10^{-3}	$3.32°$	3.95×10^{-4}	$0.05°$	1.18×10^{-4}	$0.03°$
0.50% non-uniform noise	8.02×10^{-6}	$0.01°$	0.044	$5.70°$	2.39×10^{-5}	$0.01°$
50% non-uniform simplification	0.40	$82.53°$	2.30×10^{-3}	$0.18°$	5.51×10^{-4}	$0.05°$

The mesh center coordinates are calculated as the following moment ratios:

$$C = (x_c, y_c, z_c) = (\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}}) \tag{6}$$

The principal axes of the mesh are obtained as the ordered eigenvectors of the following matrix:

$$M = \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix} \tag{7}$$

In our implementation, the most significant principal axis is aligned with axis Z. In order to resolve the axis alignment ambiguity, besides the compliance to the right-hand rule, we impose some other geometric constraints (e.g. the global moments m_{300} and m_{030} should be positive after alignment [20]). Note that the obtained normalized mesh has null m_{100}, m_{010}, m_{001}, m_{110}, m_{101} and m_{011} moments.

The above normalization relies on the analytic volume moments and therefore is processed in a continuous space. So far, existing watermarking methods have based their normalization only on the vertex coordinates while completely discarding the mesh connectivity information [18], [19], [25]. This kind of "*discrete*" moment is not very robust, especially against non-uniform connectivity changes. Recently, Rondao-Alface et al. [26] have calculated the mesh center as the weighted average coordinates of the vertices, which is equivalent to the calculation based on the mesh *surface* moments [21]. Table I presents the robustness evaluation results of the normalizations based on discrete, surface and volume moments, in terms of the center norm $\|C\|$ and the maximum principal axis (PA) variations. The experiments are carried on the Venus head mesh illustrated in Fig. 6.(a) that owns 100759 vertices. The volume moments present the best overall performances, especially under spatially non-uniform noise addition and simplification.

B. Decomposing the Mesh into Patches

The mesh is then decomposed into patches so as to insert one bit per patch. After the normalization, each vertex $v_k = (x_k, y_k, z_k)$ is converted into cylindric coordinate system as $v_k = (h_k, r_k, \theta_k) = (z_k, \sqrt{x_k^2 + y_k^2}, tan^{-1}(\frac{y_k}{x_k}))$. The patch decomposition is simply a uniform discretization of the h and θ domains into I_h and I_θ intervals with two steps h_{step} and θ_{step}. This discretization may be pseudo-randomized by using a secret key in order to still reinforce the watermarking security.

Each vertex is associated to its proper patch by calculating its discretized indices $ind(h_k)$ and $ind(\theta_k)$; however some facets may cover several patches. These facets have to be split into several smaller ones, each of which completely lies in a single patch. This facet split process is necessary to ensure a precise patch moment calculation, which is critical for the watermark robustness. The task is accomplished by automatically adding auxiliary vertices and edges on the patch borders [see Fig. 2.(c)(d)]. The whole decomposition process can be considered as a segmentation of the mesh by intersecting some 3D planes with the mesh surface in a continuous space. The mesh is now decomposed into $I_h \times I_\theta$ patches $\mathcal{P}_{j,j=0,1,...,I_h.I_\theta-1}$. These patches are ordered according to their spatial positions and their indices are determined as $j = ind(h_k).I_\theta + ind(\theta_k)$.

I_h and I_θ are two important parameters for our algorithm: if we increase the patch number, the watermark *capacity* (i.e. the watermark bit number) is increased, but it will experimentally introduce higher-amplitude patch deformation if a comparable robustness level is required, and visible distortions are prone to occur. The explanation is as follows: when the mesh is decomposed into a high number of patches (imagine the extreme case where each

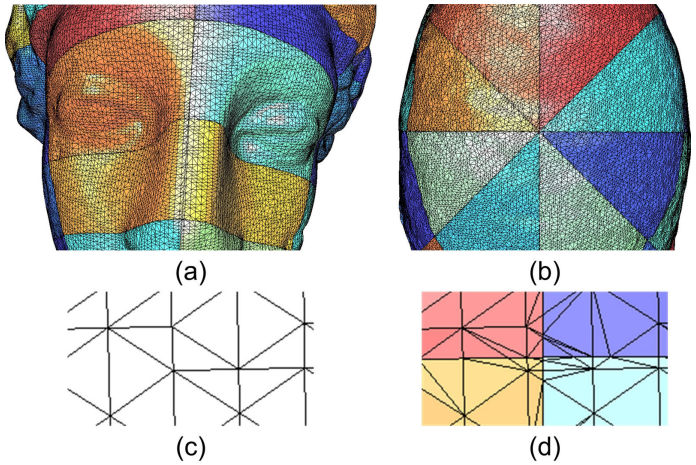

Fig. 2. (a), (b) and (d) illustrate three close-ups of the decomposed Venus head mesh; the original connectivity of (d) is shown in (c).

patch contains just one vertex), the final deformation will become of high frequency, which is more visible and less robust. $I_h = 11$ and $I_\theta = 8$ seems to achieve a good trade-off between watermark capacity, robustness and imperceptibility for most meshes. Moreover, it allows to insert around 64 bits, which is a common and sufficient payload for a robust readable watermark used for copyright protection. An adaptable setting of these two parameters according to the mesh shape constitutes one part of our future work.

The combination "mesh normalization + cylindric discretization" constitutes a simple but effective mesh decomposition. First, it can reproduce exactly the same decomposition at extraction in a blind way, with an intrinsic patch order. Besides, this decomposition depends only on the center and the principal axes of the object and is also very robust to geometry and connectivity attacks; we think that a too much shape-dependent decomposition algorithm (e.g. based on curvature or Reeb graph) would have not been so robust to various distortions. Also note that even if the cylindric decomposition is not one-to-one (e.g. multiple layers of the object may have the same $h-\theta$ range), the mesh can still be robustly decomposed as long as it is orientable; indeed we don't want to create a real mapping such as in mesh parametrization [27]. The stability of the decomposition (i.e. normalization + discretization) has been justified by the stability of the patch zero-order moment values under different attacks, even non-uniform [see Fig. 3 for the results on the watermarked Horse illustrated in Fig. 7.(b) that owns 112642 vertices]. These results also demonstrate the robustness of these local volume moments, as well as the interest of using them as watermarking primitives. The decomposition is not robust against strong local deformation and cropping, which are quite difficult to handle for blind watermarking methods. These attacks cause serious desynchronization problem to our method due to the deviation of the mesh normalization. Our normalization also fails for spheres and some other special objects, for which it is difficult to estimate the principal axes; however, most existing mesh segmentation methods would also fail to decompose consistently a sphere object. Moreover, in real life, this kind of n-symmetric object remains marginal.

C. Patch Classification and Watermark Synchronization

The obtained patches are classified into three groups:

1) *cover* patches for watermark bit insertion;
2) *discarded* patches not suitable for deformation;
3) *compensation* patches for moment compensation.

The discarded patches will not be used for bit insertion nor for moment compensation. They include the *small* patches having very low zero-order moment amplitudes, the *flat* patches having very small h domain ranges, and the *narrow* patches having very small θ domain ranges. It is in practice difficult to deform these singular patches equally strongly as the other patches, and the bits hidden in them are often less robust; thus they are discarded. Three empirical thresholds $\bar{m_{000}} = 0.0005$ for zero-order moment, $\bar{h_r} = 0.35 \times h_{step}$ for h domain range, and $\bar{\theta_r} = 0.35 \times \theta_{step}$ for θ domain range have been set to filter out these patches.

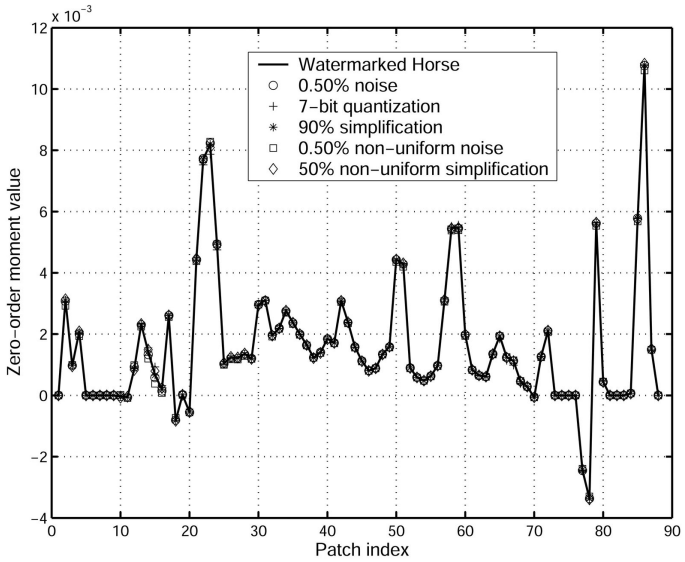

Fig. 3. Stability of the patch moment values of the watermarked Horse under different attacks.

The compensation patches serve to be deformed after watermark bit insertion so as to recover the mesh center position and principal axis orientations. The patches with larger moment amplitudes are favorable for this task since they allow a larger moment variation while keeping the deformation imperceptible. The 12 patches with the largest m_{000} amplitudes are kept from watermark bit insertion and considered as compensation patches. They are noted as $\mathcal{P}^c_{l,l=0,1,..,11}$. A compensation patch with a smaller index in this sequence has a larger m_{000} amplitude.

All the other N patches are cover patches and are denoted by $\mathcal{P}^w_{n,n=0,1,..,N-1}$. A cover patch with smaller index in this sequence also has smaller index in the global indexing $\mathcal{P}_{j,j=0,1,...,I_h.I_\theta-1}$. This cover patch order is used for the watermark synchronization: watermark bits are sequentially inserted in these ordered cover patches.

The above patch classification may introduce causality and desynchronization problems. For instance, after the watermark insertion or an attack, a compensation patch can become a cover patch (if its m_{000} amplitude decreases). Hence, special cares are taken for the potentially sensitive patches during the watermark insertion (e.g. the m_{000} amplitude of the compensation patch $\mathcal{P}^c_{11}$ is constrained to be increased), in order to preserve and reinforce the established classification. Experimentally, the desynchronization never happens under week and moderate attacks, and also rarely occurs under strong attacks. Our current solution is to realize several different extractions (normally less than 4 even under very strong attacks) by classifying the suspicious patch(es) in different possible groups. The desynchronization would also be resolved by transmitting an additional sequence of $I_h \times I_\theta$ bits at the extraction side to explicitly indicate the cover patch locations; however, strictly speaking, this solution would make the algorithm semi-blind.

D. Patch Moment Quantization

After patch decomposition and classification, the mesh allows for the insertion of $(N-1)$ bits $w_1, w_2, ..., w_{N-1}$ in its N cover patches (the first cover patch is not watermarked, see Eq. 9). The watermark bit $w_n \in \{0,1\}$ is inserted by quantizing the zero-order moment $m_{000}^{(\mathcal{P}^w_n)}$. The proposed quantization scheme is a modified version of the scalar Costa scheme (SCS) [28] that is widely used in image, audio and video watermarking. The basic idea of SCS is that instead of replacing the initial value exactly by a quantized value, it is better to push the initial value towards the quantized one. SCS provides more control on the insertion intensity and offers a higher security level compared to the simple substitutive quantization.

The practical quantization procedure is as follows: first, a component-wise pseudo-random codebook is established for each $m_{000}^{(\mathcal{P}^w_n)}$ as given by Eq. 8, where $\Delta^{(\mathcal{P}^w_n)}$ is the quantization step, $z \in \mathcal{Z}$ is an integer, $a \in \{0,1\}$ stands for the implied bit of a codeword u, and $t^{(\mathcal{P}^w_n)}\Delta^{(\mathcal{P}^w_n)}$ is an additive pseudo-random dither signal.

$$\mathcal{U}_{m_{000}^{(\mathcal{P}^w_n)},t^{(\mathcal{P}^w_n)}} = \bigcup_{a=0}^{1} \left\{ u = z.\Delta^{(\mathcal{P}^w_n)} + a\frac{\Delta^{(\mathcal{P}^w_n)}}{2} + t^{(\mathcal{P}^w_n)}\Delta^{(\mathcal{P}^w_n)} \right\} \tag{8}$$

In our implementation, $t^{(\mathcal{P}_n^w)}, 1 \leq n < N$ form a simulation sequence of a random variable T uniformly distributed between $\left[-\frac{1}{2}, \frac{1}{2}\right]$ and are generated by inputting a secret key K into an appropriate pseudo-random number generator.

Differently from in [28], the quantization step $\Delta^{(\mathcal{P}_n^w)}$ in our case is no longer fixed and is also component-wise. A fixed step, even combined with an adaptive compensation factor value (introduced in Eq. 11), is experimentally not appropriate, since different patches can tolerate different moment variations. There are always difficulties in introducing quantization-based schemes for mesh watermarking, even to quantize directly the vertex coordinates. For instance, a coarse mesh can tolerate a larger step than a dense mesh, and a rough region can be deformed much stronger than a smooth region without being noticed. We propose the derivation of $\Delta^{(\mathcal{P}_n^w)}, 1 \leq n < N$ as follows:

$$\Delta^{(\mathcal{P}_n^w)} = \begin{cases} \Delta_{pre} \cdot \left| m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)} / \left\lceil \frac{m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}}{m_{000}^{(\mathcal{P}_n^w)}} \right\rceil \right|, & \text{if } \left| \frac{m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}}{m_{000}^{(\mathcal{P}_n^w)}} \right| > 1 \\ \Delta_{pre} \cdot \left| m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)} \cdot \left\lfloor \frac{m_{000}^{(\mathcal{P}_n^w)}}{m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}} \right\rfloor \right|, & \text{if else} \end{cases} \tag{9}$$

where $m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}$ is the watermarked moment value of the patch $\mathcal{P}_{n-1}^w$ with $m_{000}^{(\hat{\mathcal{P}}_0^w)} = m_{000}^{(\mathcal{P}_0^w)}$, and Δ_{pre} is given by:

$$\Delta_{pre} = \begin{cases} 0.04, & \text{if } \left| m_{000}^{(\mathcal{P}_n^w)} \right| > 0.01 \\ 0.07, & \text{if } m_{000}^{-} < \left| m_{000}^{(\mathcal{P}_n^w)} \right| \leq 0.01 \end{cases} \tag{10}$$

We can notice that the quantization step of the current patch $\mathcal{P}_n^w$ is related to the quantized moment of its previous patch $\mathcal{P}_{n-1}^w$. This is inspired from the work of Pérez-González et al. [29]. Their rational dither modulation (RDH) method achieves the invariance to value-metric scaling attacks for the quantization index modulation paradigm [30]. We have proposed the above RDH-like scheme in order to reinforce the watermark robustness against the alteration of the farthest vertex that is used for mesh scaling during the normalization step (see Section IV-A). This alteration is possible during the watermark insertion and after attacks. The introduction of $m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}$ in the calculation of $\Delta^{(\mathcal{P}_n^w)}$ makes the quantization scheme intrinsically invariant to scaling thus can effectively handle the local scaling phenomenon caused by this farthest vertex alteration.

The quantization step $\Delta^{(\mathcal{P}_n^w)}$ calculated using Eqs. 9 and 10 is approximately proportional to its moment amplitude $\left| m_{000}^{(\mathcal{P}_n^w)} \right|$ so that the patches with larger moment amplitudes can adaptively have larger moment variations. There are different Δ_{pre} values for patches having moderate moment amplitudes and those having large amplitudes (Eq. 10). This distinction helps to balance the induced distortions on these different patches and is also theoretically reasonable (please refer to Proof 1 in the support document). Although a more sophisticated derivation of Δ_{pre} may be possible, the empirical setting in Eq. 10 has already worked well enough in practice for many meshes.

Once the codebook is constructed, we find the nearest codeword $u_{m_{000}^{(\mathcal{P}_n^w)}}$ to $m_{000}^{(\mathcal{P}_n^w)}$ that correctly implies the watermark bit w_n (i.e. w_n should be equal to value a in the derivation of $u_{m_{000}^{(\mathcal{P}_n^w)}}$). The quantized value $m_{000}^{(\hat{\mathcal{P}}_n^w)}$ is calculated according to Eq. 11, where $\alpha^{(\mathcal{P}_n^w)} \in [0, 1]$ is a compensation factor. Normally, we take $\alpha^{(\mathcal{P}_n^w)} \geq 0.50$ in order to ensure the correctness of the watermark extraction when there is no attack.

$$m_{000}^{(\hat{\mathcal{P}}_n^w)} = m_{000}^{(\mathcal{P}_n^w)} + \alpha^{(\mathcal{P}_n^w)} (u_{m_{000}^{(\mathcal{P}_n^w)}} - m_{000}^{(\mathcal{P}_n^w)}) \tag{11}$$

$\alpha^{(\mathcal{P}_n^w)}$ will partially drive the induced distortion and the watermarking security. A perfect security of the classical 2-symbol SCS quantization (i.e. the secrecy of K) can be gained if $\alpha^{(\mathcal{P}_n^w)} = 0.50$ for all the patches [31]; then, the information leakage increases as $\alpha^{(\mathcal{P}_n^w)}$ increases. Although our scheme is slightly different from SCS, we still follow this principle, trying to keep $\alpha^{(\mathcal{P}_n^w)}$ close to 0.50.

It is possible that, after quantization, the ceiled ($\lceil\ \rceil$) or floored ($\lfloor\ \rfloor$) integer moment ratio between $m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}$ and $m_{000}^{(\hat{\mathcal{P}}_n^w)}$ (see Eq. 9) may be different from that between $m_{000}^{(\hat{\mathcal{P}}_{n-1}^w)}$ and $m_{000}^{(\mathcal{P}_n^w)}$, so that the quantization step can be different at extraction. For the sensitive ratios close to integers, which in practice rarely occur, we automatically adjust the corresponding compensation factors so that these integer moment ratios are kept unchanged after quantization; however, the correctness of the bit insertion cannot always be ensured.

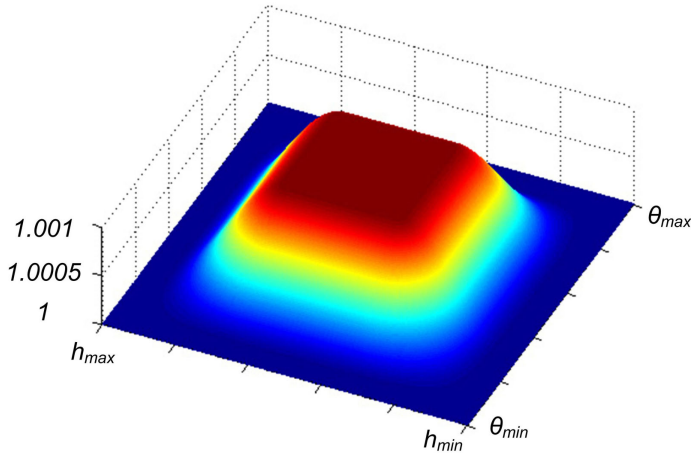

Fig. 4. Illustration of the deformation mask function pattern. Here the global deformation factor s is equal to 1.001.

E. Patch Deformation

The next step is to deform the cover patches in order to reach the quantized target moment values. Since the volume moment transformation is not reversible, we need to modify the moment of a patch heuristically by moving its vertices. The amplitude and direction of this patch deformation is iteratively adjusted so that the patch zero-order moment gradually achieves its target value. Besides, the displacements of all the vertices within a patch are modulated by using a smooth deformation pattern function that is illustrated in Fig. 4, so that the patch's global deformation is of low frequency. Each vertex has its own multiplicative deformation factor derived from this mask function. For a vertex v_k within $\mathcal{P}_n^w$, the derivation begins with a normalization of its coordinates:

$$h_k^{'} = 1 - \left| \frac{2(h_k - h_{min}^{(\mathcal{P}_n^w)})}{h_{max}^{(\mathcal{P}_n^w)} - h_{min}^{(\mathcal{P}_n^w)}} - 1 \right| \in [0,1] \tag{12}$$

$$\theta_k^{'} = 1 - \left| \frac{2(\theta_k - \theta_{min}^{(\mathcal{P}_n^w)})}{\theta_{max}^{(\mathcal{P}_n^w)} - \theta_{min}^{(\mathcal{P}_n^w)}} - 1 \right| \in [0,1] \tag{13}$$

Under this normalization, the vertices close to the patch borders will have small $h_k^{'}$ and $\theta_k^{'}$ values, while the vertices close to the patch center will receive large values. For each vertex, two weights are then calculated: Eq. 14 gives the formula for the h domain weight calculation, the calculation of the θ domain weight $wt_{\theta_k^{'}}$ has a similar form.

$$wt_{h_k^{'}} = \begin{cases} 0 & \text{if } 0 \leq h_k^{'} < 0.1 \\ \frac{1}{2}\sqrt{|s-1|}[\sin(\frac{5\pi}{3}(h_k^{'} - \frac{2}{5})) + 1] & \text{if } 0.1 \leq h_k^{'} < 0.7 \\ \sqrt{|s-1|} & \text{if } 0.7 \leq h_k^{'} \leq 1.0 \end{cases} \tag{14}$$

where s is called the global deformation factor. The individual deformation factor s_{v_k} for vertex v_k is then determined as follows:

$$s_{v_k} = \begin{cases} 1 + wt_{h_k^{'}}.wt_{\theta_k^{'}} & \text{if } s > 1 \\ 1 - wt_{h_k^{'}}.wt_{\theta_k^{'}} & \text{if } s < 1 \end{cases} \tag{15}$$

Finally, the coordinates of a candidate moved vertex are obtained as the multiplication of its original coordinates with s_{v_k} or $(2 - s_{v_k})$, depending on the contribution sign of its incident facets (more details in Algorithm 1).

The defined deformation mask (in continuous setting, see Fig. 4) is very smooth: it is constant in the border and center regions, and has a sinus-like shape between the above two regions. Its amplitude and direction depend on the global deformation factor s. The objective now is to find, for each patch, the correct value for s that produces the target quantized moment value when applied on the original patch. For this task we have defined a simple and efficient iterative process that is summarized as Algorithm 1. Note that some vertices are not modifiable: the added border vertices, their direct neighbors, and the vertices having simultaneously facets with positive and negative

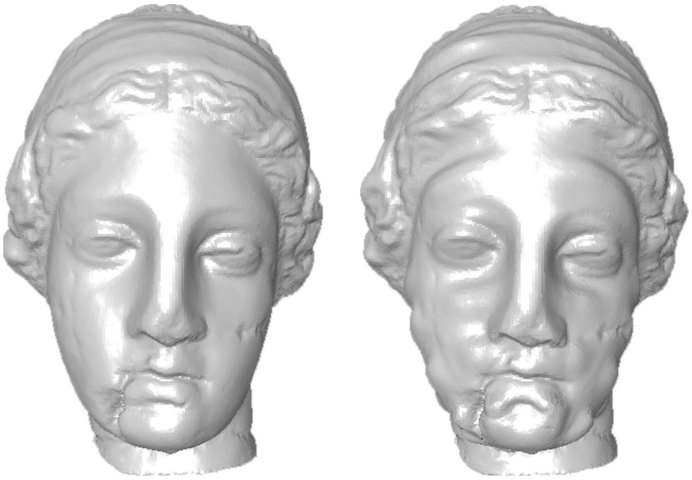

Fig. 5. A moderately watermarked Venus head (a) and a strongly watermarked one (b).

moment contributions. We have also constrained that a modified vertex cannot get out of its original patch. This iterative algorithm permits reaching the target global deformation factor s within normally less than 25 iterations. Actually, each patch can have its own mask function (e.g. the different ranges of $h_k^{'}$ in Eq. 14 can be changed); however, a uniform setting of the above mask for all the patches is already satisfying in practice.

Algorithm 1 Iterative patch deformation algorithm

Notations: global deformation factor s; its modification step k_s; obtained zero-order moment after i-th iteration m_i; original moment $m_{000}^{(\mathcal{P}_n^w)}$; target moment $m_{000}^{(\hat{\mathcal{P}}_n^w)}$

1: Determine the involved vertices for the current patch $\mathcal{P}_n^w$; for each involved vertex deduce its modifiability; for each modifiable vertex v_k record its coordinates (x_k, y_k, z_k)
2: Initialize the parameters: $s = 1$, $k_s = 0.01$, $i = 1$, $\mathrm{m}_{-1} = \mathrm{m}_0 = m_{000}^{(\mathcal{P}_n^w)}$
3: **repeat**
4: Modify s according to the following rule
- if $\mathrm{m}_{i-1} < m_{000}^{(\hat{\mathcal{P}}_n^w)}$ and $\mathrm{m}_{i-2} < m_{000}^{(\hat{\mathcal{P}}_n^w)}$, then $s \leftarrow s + k_s$;
- if $\mathrm{m}_{i-1} < m_{000}^{(\hat{\mathcal{P}}_n^w)}$ and $\mathrm{m}_{i-2} > m_{000}^{(\hat{\mathcal{P}}_n^w)}$, then $k_s \leftarrow k_s/2$ and $s \leftarrow s + k_s$;
- if $\mathrm{m}_{i-1} > m_{000}^{(\hat{\mathcal{P}}_n^w)}$ and $\mathrm{m}_{i-2} > m_{000}^{(\hat{\mathcal{P}}_n^w)}$, then $s \leftarrow s - k_s$;
- if $\mathrm{m}_{i-1} > m_{000}^{(\hat{\mathcal{P}}_n^w)}$ and $\mathrm{m}_{i-2} < m_{000}^{(\hat{\mathcal{P}}_n^w)}$, then $k_s \leftarrow k_s/2$ and $s \leftarrow s - k_s$.

5: **for** each modifiable involved vertex v_k in $\mathcal{P}_n^w$ **do**
6: Calculate v_k's deformation factor s_{v_k} (Eqs. 14, 15) according to s and its normalized coordinates (Eqs. 12, 13)
7: Modify v_k's original coordinates to obtain a candidate moved vertex $v_k^{'}$ by using the following rule
- if all v_k's incident facets have positive moment contributions, then $(x_k^{'}, y_k^{'}, z_k^{'}) = s_{v_k}.(x_k, y_k, z_k)$
- if all v_k's incident facets have negative moment contributions, then $(x_k, y_k, z_k) = (2 - s_{v_k}).(x_k, y_k, z_k)$

8: **end for**
9: evaluate m_i as the zero-order moment of the temporary deformed patch
10: iteration number incrementation: $i = i + 1$
11: **until** $\left|\mathrm{m}_i - m_{000}^{(\hat{\mathcal{P}}_n^w)}\right| < \epsilon$ or $i = I_{max}$

Figure 5 shows the distortion effects of a moderate-intensity watermark and a very strong-intensity watermark. There exists hardly any visual distortion for the former because the modification is of low frequency; for the latter, the distortion becomes visible and has a similar shape as the deformation mask.

F. Moment Compensation

The objective is to recover the mesh's center position and principal axis orientations after the deformation of the cover patches. Concretely, it needs to compensate for the variations of the mesh's m_{100}, m_{010}, m_{001}, m_{110}, m_{101} and m_{011} moments that have been caused by the watermark bit insertion, so that they become null again, or at least reasonably small. Our compensation method is based on the following result: when deforming a patch with the above iterative algorithm and the proposed deformation mask function, it can be proven (please refer to Proof 2 in the support document) and has also been experimentally validated that the following moment variation ratios are approximately constant under different s values: $\frac{\Delta m_{100}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$, $\frac{\Delta m_{010}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$, $\frac{\Delta m_{001}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$, $\frac{\Delta m_{110}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$, $\frac{\Delta m_{101}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$ and $\frac{\Delta m_{011}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$. The compensation patches are deformed arbitrarily by using Algorithm 1 prior to the compensation step in order to learn these ratios (of course these patches are then restored before the compensation). For the notation simplicity, the six learned ratios of $\mathcal{P}_l^c$ are hereafter denoted by $\mathbf{r}_1^l$ to $\mathbf{r}_6^l$.

The problem is then formulated as the deduction of the correct moment variations Δm_{000}^l for the 12 compensation patches such that the variations of the other moments compensate for the global moments $\tilde{m}_{100}$, $\tilde{m}_{010}$, $\tilde{m}_{001}$, $\tilde{m}_{110}$, $\tilde{m}_{101}$ and $\tilde{m}_{011}$ of the obtained mesh after bit insertion. A 6×12 linear least-squares system is constructed:

$$M = \arg\min \left\| R.M - \tilde{M} \right\|_2^2 \tag{16}$$

where R is a 6×12 matrix with $R_{ij} = \mathbf{r}_i^{j-1}$, M is a 12×1 matrix with $M_{i1} = \Delta m_{000}^{i-1}$, and $\tilde{M} = [\tilde{m}_{100}\ \ \tilde{m}_{010}\ \ \tilde{m}_{001}\ \ \tilde{m}_{110}\ \ \tilde{m}_{101}$ The optimization of the above system is subject to two constraints:

$$Lb \leq M \leq Ub \tag{17}$$

$$R^{'}.M^{'} = \tilde{M}^{'} \tag{18}$$

where Lb and Ub prescribe the lower and upper bounds of the moment variations, and $R^{'}$, $M^{'}$ and $\tilde{M}^{'}$ are composed of the last three rows of R, M and $\tilde{M}$, respectively. The first constraint is related to the deformation imperceptibility. Practically, we have set the moment variation lower and upper bounds so that the deformation of the compensation patches is the same order as that of the cover patches. The second constraint defines the priority of compensating the second-order moments. The introduction of this second constraint is based on the fact that our whole watermarking algorithm is experimentally much more sensitive to the principal axis change than to the mesh center change.

We then resolve the system and deduce the moment variations (i.e. the target moment values) for the compensation patches; the corresponding deformation is achieved by using Algorithm 1. After this step, the six compensated first and second order moments of the watermarked mesh are very close to zeros and have no longer any negative influence on the blind watermark extraction. The last step of the watermark insertion is the removal of all the inserted auxiliary vertices and edges.

V. Watermark Extraction

The watermark extraction is blind and fast. First, the input mesh is normalized by using the technique described in Section IV-A. Then, the vertex coordinates are converted into cylindric system and the mesh is decomposed into patches by discretizing its h and θ domains. After using the patch classification rules given in Section IV-C, we can select out the candidate cover patches for bit extraction. Next, with the knowledge of the secret key K and by using Eqs. 8 to 10, we construct a codebook $\hat{\mathcal{U}}_{m_{000}^{(\mathcal{P}_n^w)},t^{(\mathcal{P}_n^w)}}$ for each cover patch. According to the actual moment value $m_{000}^{(\breve{\mathcal{P}}_n^w)}$ of the patch, we can find the codeword $\hat{u}_{m_{000}^{(\breve{\mathcal{P}}_n^w)}}$ that is the closest to $m_{000}^{(\breve{\mathcal{P}}_n^w)}$ in the constructed codebook. Finally, the n-th extracted watermark bit $w_n^{'}$ is considered as the implied bit a of the codeword $\hat{u}_{m_{000}^{(\breve{\mathcal{P}}_n^w)}}$.

VI. Experimental Results

A. Basic simulations

The proposed method has been tested on several meshes. Figure 6 illustrates four of them: Venus (100759 vertices), Horse (112642 vertices), Bunny (34835 vertices) and Dragon (50000 vertices). The adjustable parameters of our algorithm are the compensation factors $\alpha^{(\mathcal{P}_n^w)}$ for the cover patches, which drive the trade-off between distortion, robustness and security. They have been fixed for the different meshes (see Table II) by observing two

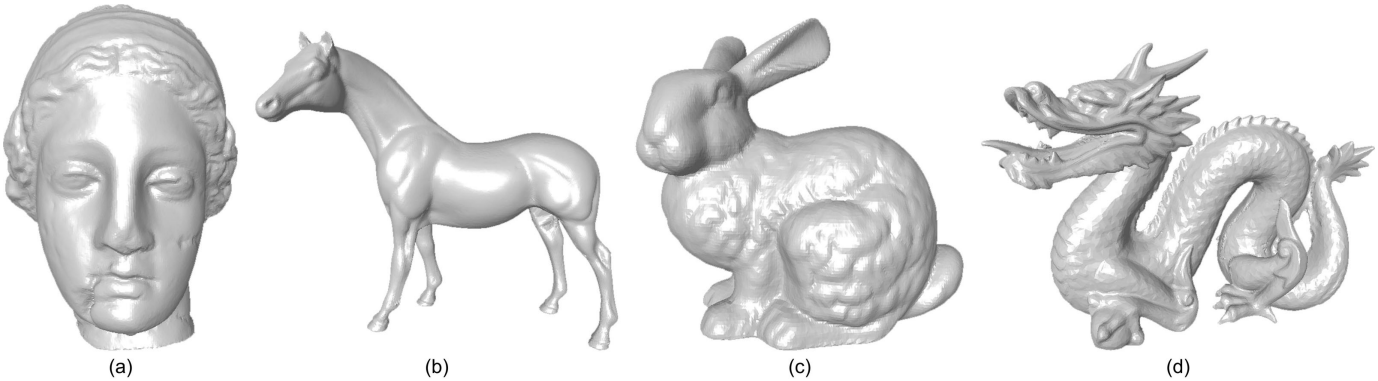

Fig. 6. The original non-watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, (d) Dragon.

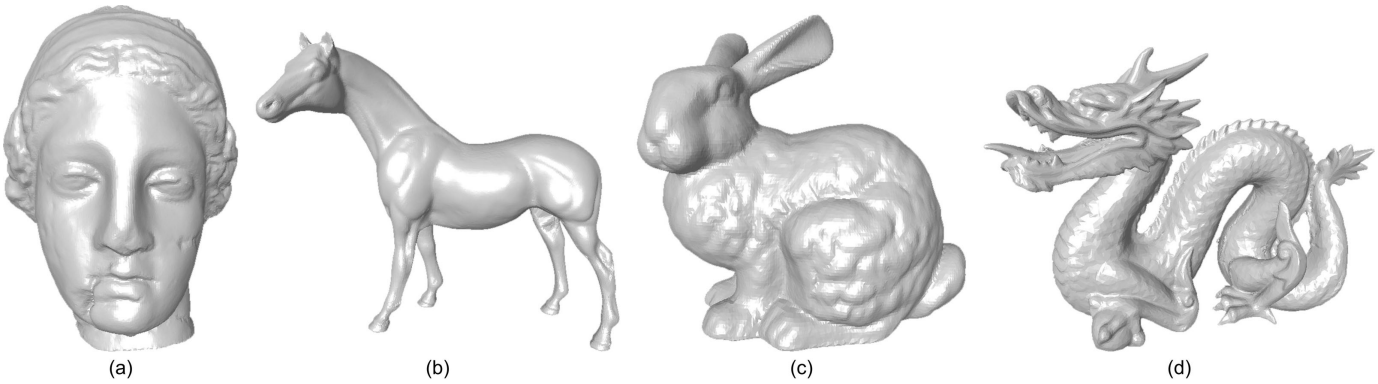

Fig. 7. The watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, (d) Dragon.

empirical rules: (1) they cannot be too large in reason of security, and (2) the meshes having lower sampling density can support larger values since a stronger deformation can be introduced on them without being noticed. Figure 7 illustrates the watermarked meshes; compared with Fig. 6, there exist nearly no perceptible distortions introduced by the watermark embedding, even on very smooth regions such as the body of the Horse. The main reason is that these induced distortions are smooth and of low frequency, to which the human eyes are not very sensitive [5]. Figure 8 illustrates the corresponding geometric objective distortion maps; we can notice that although the distortion is globally well balanced, there still exist some patches that are much more deformed than others.

Table II details some statistics about the watermark insertion and extraction. All the tests have been carried out on a Pentium IV 2.0GHz processor with 2GB memory. The objective distortions between the normalized watermarked and original meshes are measured by Metro [32] in terms of maximum root mean square error (MRMS). A "perceptual" distance between them is evaluated by the mesh structural distortion measure (MSDM) proposed in [33]. Its value tends towards 1 (theoretical limit) when the measured objects are visually very different and is equal

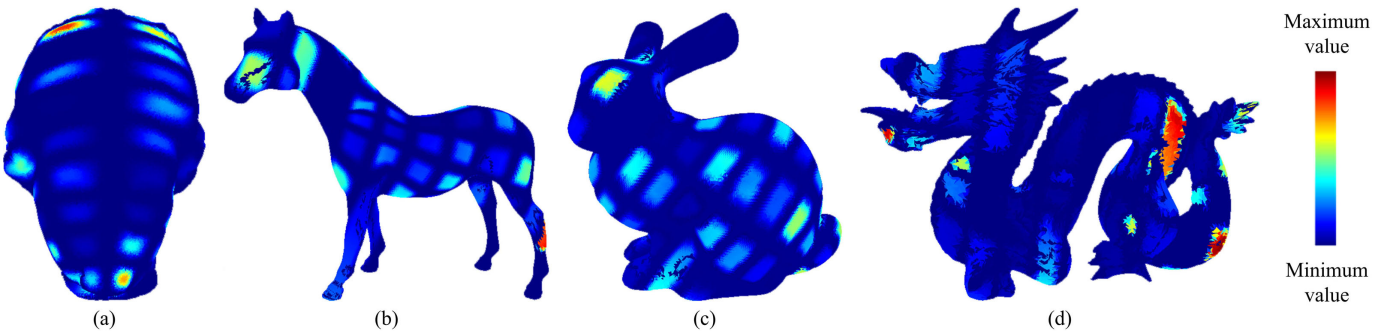

Fig. 8. The objective distortion maps of the watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, (d) Dragon.

TABLE II
BASELINE EVALUATIONS OF THE PROPOSED METHOD

	Venus	Horse	Bunny	Dragon
$\alpha^{(\mathcal{P}_n^w)}$	0.70	0.75	0.80	0.85
Embedding time (s)	410.8	191.5	109.4	166.2
Extraction time (s)	3.2	2.9	1.1	1.6
WM capacity (bit)	75	46	67	49
BER when no attack	0.03	0	0	0
MRMS by WM (10^{-3})	2.34	1.04	1.75	1.76
MSDM by WM	0.15	0.17	0.19	0.20

to 0 for identical ones. One advantage of our method is that it can introduce relatively high-amplitude deformation while keeping it imperceptible. The nonzero bit error rate (BER) of the extraction without attack on Venus is due to the moment ratio sensitivity problem mentioned at the end of Section IV-D. Most of the embedding time is spent on the iterative deformation step, which does not only depend on the mesh size (i.e. its vertex number), but also on its cover patch number. The extraction time is almost completely due to the patch decomposition and is basically proportional to the mesh size.

B. Robustness against Geometry Attacks

In the following subsections, the resistance of the inserted watermark is tested under different types of attacks. The robustness is evaluated by the BER (bit error rate) and the normalized correlation [1] (given by Eq. 19) between the extracted watermark bit string $\{w'_n\}$ and the originally inserted one $\{w_n\}$.

$$Corr. = \frac{\sum_{n=1}^{N-1}(w'_n - \bar{w'_n})(w_n - \bar{w_n})}{\sqrt{\sum_{n=1}^{N-1}(w'_n - \bar{w'_n})^2 . \sum_{n=1}^{N-1}(w_n - \bar{w_n})^2}} \tag{19}$$

where $\bar{w'_n}$ and $\bar{w_n}$ indicate the averages of the watermark bits. This correlation value varies between -1 (orthogonal strings) and $+1$ (the same strings). The distortions induced by the attacks are measured by MRMS. This subsection presents the test results under geometry attacks.

Our watermark is experimentally perfectly invariant to the so-called *content preserving attacks* including vertex/facet reordering in the mesh file and similarity transformations (i.e. translation, rotation, uniform scaling and their combination). Tables III, IV and V respectively present the robustness evaluations against noise addition, smoothing and uniform coordinate quantization. Some geometrically attacked models are illustrated in Fig. 9.(a)-(d). The maximum amplitude of the random additive noise is relative to the average distance from the vertices to the mesh center. The noise intensity of each vertex is uniformly distributed between 0 and the maximum amplitude. For each amplitude, we perform five experiments using different seeds to generate different noise patterns and report the average as the final result. For spatially non-uniform noise addition, a random and sufficient part of the mesh is noised while keeping the other part untouched. For smoothing attacks, we have considered a Laplacian smoothing [34] with different iteration numbers while fixing the deformation factor λ as 0.03. Our algorithm demonstrates a fairly high robustness against geometry attacks, even with relatively strong amplitudes or non-uniformity. For instance, in average, we can still successfully extract up to 87% of the mark under 0.50% noise addition [see Fig. 9.(a)]. The Bunny and Dragon are less robust to smoothing because this attack produces an important shrinking effect on these two models.

C. Robustness against Connectivity Attacks

The tested connectivity attacks include simplification (uniform and non-uniform), subdivision and remeshing. The used mesh simplification algorithm is the one based on quadric error metrics proposed by Garland and Heckbert [35]. The subdivision attacks include the simple midpoint scheme, the modified butterfly scheme and the Loop scheme [36]. The remeshing attack is a uniform resampling of the mesh vertices using ReMESH [37]; two different target vertex numbers are considered for this resampling: 100% and 50% of the original vertex number. Tables VI, VII and VIII present the robustness evaluations against these attacks. In Fig. 9.(e)-(h), some attacked models are illustrated. It can be observed that our scheme owns a very high robustness against all these attacks, which are

TABLE III
ROBUSTNESS AGAINST RANDOM NOISE ADDITION

Model	Amplitude	MRMS (10^{-3})	BER	Corr.
Venus	0.10%	0.33	0.03	0.94
	0.30%	0.98	0.06	0.87
	0.50%	1.63	0.11	0.78
	non-unif. 0.30%	0.68	0.05	0.89
	non-unif. 0.50%	1.13	0.13	0.73
Horse	0.10%	0.21	0.01	0.98
	0.30%	0.64	0.08	0.86
	0.50%	1.07	0.12	0.77
	non-unif. 0.30%	0.45	0.04	0.92
	non-unif. 0.50%	0.78	0.11	0.78
Bunny	0.10%	0.22	0.01	0.98
	0.30%	0.66	0.07	0.85
	0.50%	1.11	0.11	0.77
	non-unif. 0.30%	0.50	0.02	0.95
	non-unif. 0.50%	0.82	0.07	0.85
Dragon	0.10%	0.24	0.01	0.98
	0.30%	0.72	0.12	0.76
	0.50%	1.20	0.19	0.61
	non-unif. 0.30%	0.63	0.14	0.72
	non-unif. 0.50%	0.94	0.24	0.53

TABLE IV
ROBUSTNESS AGAINST LAPLACIAN SMOOTHING ($\lambda = 0.03$)

Model	Iteration	MRMS (10^{-3})	BER	Corr.
Venus	10	0.12	0.04	0.92
	50	0.51	0.04	0.92
	100	0.88	0.08	0.84
Horse	10	0.07	0	1
	50	0.29	0.07	0.87
	100	0.52	0.13	0.74
Bunny	10	0.26	0.13	0.73
	30	0.69	0.19	0.62
	50	1.04	0.37	0.27
Dragon	10	0.31	0.08	0.84
	30	0.82	0.24	0.52
	50	1.28	0.41	0.19

TABLE V
ROBUSTNESS AGAINST UNIFORM COORDINATE QUANTIZATION

Model	Intensity	MRMS (10^{-3})	BER	Corr.
Venus	9-bit	0.66	0.04	0.92
	8-bit	1.32	0.11	0.81
	7-bit	2.70	0.11	0.79
Horse	9-bit	0.49	0	1
	8-bit	0.97	0.15	0.70
	7-bit	2.05	0.26	0.49
Bunny	9-bit	0.52	0.04	0.91
	8-bit	1.05	0.04	0.91
	7-bit	2.07	0.15	0.70
Dragon	9-bit	0.57	0.02	0.96
	8-bit	1.13	0.18	0.63
	7-bit	2.29	0.39	0.23

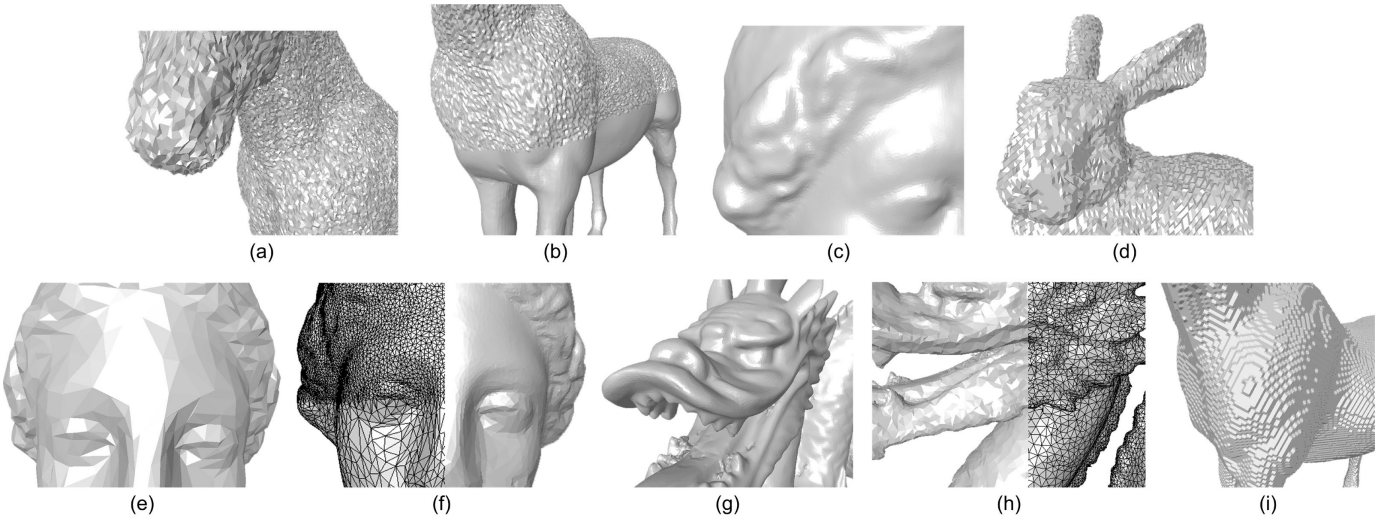

Fig. 9. Close-ups of some attacked watermarked models: (a) 0.50% random additive noise ($BER = 0.12$); (b) 0.50% spatially non-uniform noise ($BER = 0.11$); (c) 100-iteration Laplacian smoothing with $\lambda = 0.03$ ($BER = 0.08$); (d) 7-bit coordinate quantization ($BER = 0.15$); (e) uniform simplification by 97.5% vertex reduction ($BER = 0.07$); (f) non-uniform simplification by 75% vertex reduction, the upper and lower parts are simplified with different reduction ratios ($BER = 0.09$); (g) 1 Loop subdivision ($BER = 0.06$); (h) uniform remeshing with original vertex number ($BER = 0.10$); (i) output mesh of the Marching Cubes algorithm on a $350 \times 350 \times 350$ discretized Horse ($BER = 0.11$).

TABLE VI
ROBUSTNESS AGAINST SIMPLIFICATION

Model	Reduction ratio	MRMS (10^{-3})	BER	Corr.
Venus	90%	0.29	0.03	0.95
	95%	0.51	0.05	0.89
	97.5%	0.91	0.07	0.84
	non-unif. 50%	0.25	0.04	0.92
	non-unif. 75%	0.67	0.09	0.82
Horse	90%	0.13	0	1
	95%	0.24	0.02	0.96
	97.5%	0.43	0.07	0.87
	non-unif. 50%	0.21	0.09	0.83
	non-unif. 75%	0.35	0.11	0.78
Bunny	70%	0.21	0	1
	90%	0.54	0.13	0.73
	95%	0.95	0.13	0.74
	non-unif. 25%	0.17	0	1
	non-unif. 50%	0.66	0.13	0.73
Dragon	70%	0.37	0	1
	90%	1.00	0.22	0.56
	95%	1.79	0.46	0.08
	non-unif. 25%	0.23	0	1
	non-unif. 50%	0.86	0.16	0.67

considered difficult to handle for a blind mesh watermarking algorithm. As an example, for Venus and Horse, we can still retrieve 93% of the mark after having removed 97.5% of the vertices. The Dragon is less robust against these connectivity attacks since it owns a relatively low number of vertices regarding its complexity, thus modifying its connectivity induces important modifications on its shape.

D. Robustness against Representation Conversion

We have tested one scenario of this serious attack: the watermarked mesh is discretized into a $350 \times 350 \times 350$ voxel grid. To extract the watermark from this discrete volumetric representation, we transform it back into a mesh by using the well known Marching Cubes algorithm [38]. The watermark extraction is then carried out on this reconstructed mesh. Table IX presents the robustness results under this attack. For Venus, Horse and Bunny,

TABLE VII
ROBUSTNESS AGAINST ONE SUBDIVISION

Model	Scheme	MRMS (10^{-3})	BER	Corr.
Venus	Midpoint	0	0.03	0.95
	m-Butterfly	0.10	0.03	0.95
	Loop	0.11	0.04	0.92
Horse	Midpoint	0	0	1
	m-Butterfly	0.05	0	1
	Loop	0.06	0	1
Bunny	Midpoint	0	0	1
	m-Butterfly	0.23	0	1
	Loop	0.23	0.15	0.71
Dragon	Midpoint	0	0	1
	m-Butterfly	0.24	0.02	0.96
	Loop	0.25	0.06	0.88

TABLE VIII
ROBUSTNESS AGAINST UNIFORM REMESHING

Model	Vertex number	MRMS (10^{-3})	BER	Corr.
Venus	100%	0.08	0.04	0.92
	50%	0.30	0.04	0.92
Horse	100%	0.06	0	1
	50%	0.18	0.04	0.91
Bunny	100%	0.39	0.03	0.94
	50%	0.63	0.13	0.74
Dragon	100%	0.40	0.10	0.80
	50%	1.54	0.45	0.11

the robustness is very satisfying (BER is around 0.12) considering the strength of the attack [see Fig. 9.(i)]. The extraction on Dragon fails because the Marching Cubes algorithm has created very strong artefacts on its tail, which significantly changes the mesh's center and principal axes.

E. Discussion and Comparison

In this subsection, we discuss our scheme and compare it with the two recent methods from Cho et al. [19], which are considered as the most robust blind algorithms from the state-of-the-art. We have applied their algorithms on Horse (algorithm I) and Bunny (algorithm II) so as to compare the results in terms of imperceptibility and robustness.

First, concerning the watermark imperceptibility, the induced patch deformation in our scheme is of low frequency while their methods seem to produce relatively high frequency artefacts. Figure 10 illustrates the Horse and Bunny models watermarked by their and our methods. Although the introduced objective MRMS distances by their algorithms (0.51×10^{-3} for Horse and 0.29×10^{-3} for Bunny) are smaller, the induced mesh modifications are more visible. This is confirmed by the MSDM distances between their watermarked and original models, which reflect the perceptual visual difference (0.23 for Horse and 0.32 for Bunny against respectively 0.17 and 0.19 for our method). In particular, some ring-like high frequency artefacts can be perceived on their watermarked meshes, especially on smooth regions such as the body of the Horse. Indeed, in their methods, the watermark is inserted by modifying the vertex norm histogram without considering the relative spatial positions of the vertices; while during our watermark embedding, the deformation pattern of each patch is carefully controlled. Besides, it can

TABLE IX
ROBUSTNESS AGAINST VOXELIZATION

Model	Intensity	MRMS (10^{-3})	BER	Corr.
Venus	$350 \times 350 \times 350$	0.95	0.13	0.74
Horse	$350 \times 350 \times 350$	1.22	0.11	0.78
Bunny	$350 \times 350 \times 350$	0.85	0.12	0.76
Dragon	$350 \times 350 \times 350$	7.27	0.55	-0.11

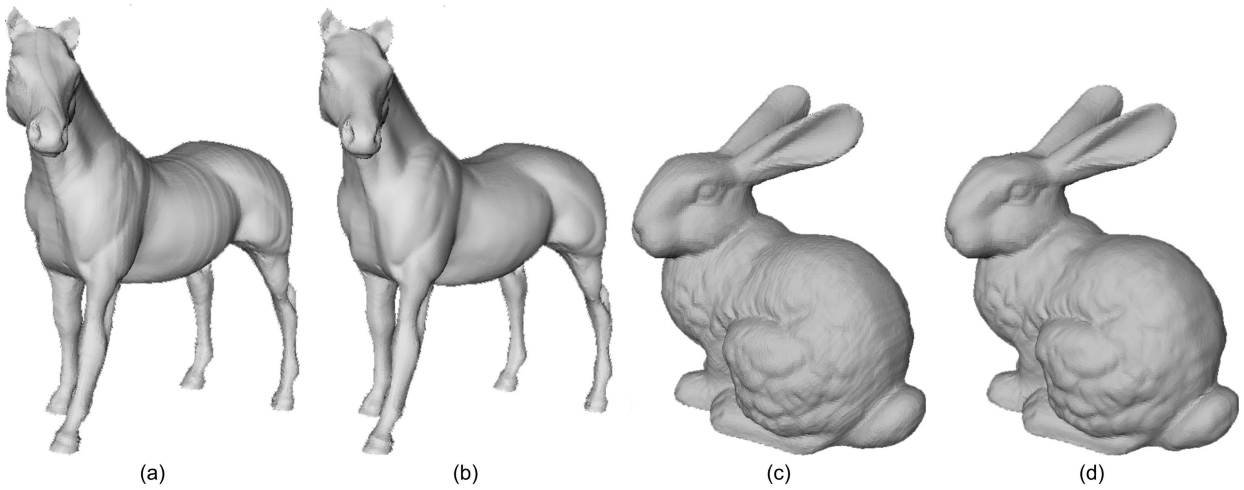

Fig. 10. Imperceptibility comparison between the algorithms of Cho et al. and our method: (a) Horse watermarked by their algorithm I (strength $\alpha = 0.03$); (b) Horse watermarked by our method; (c) Bunny watermarked by their algorithm II (strength $\alpha = 0.07$); (d) Bunny watermarked by our method.

be seen from Tables III-IX that our deformation induces a comparable MRMS distance as some strong attacks. This illustrates the effectiveness of our mask-based patch deformation algorithm and the necessity of the moment compensation post-processing.

Tables X and XI present the robustness evaluations of the methods of Cho et al. corresponding to the watermarked models illustrated in Fig. 10.(a)(c). The correlation values of our method are also listed in the tables. For a fair comparison, we have just inserted 46 bits in the Horse to ensure a same capacity for both schemes on this model. Our watermarked Horse that has nearly no visual distortions is much more robust than theirs that presents obvious noticeable distortions, under both geometry and connectivity attacks. Our algorithm is particularly more robust to quantization (our correlation is 1 against 0.66 for their algorithm, under 9-bit quantization) and simplification (1 against 0.58 for 90% simplification). Our watermarked Bunny has also a better imperceptibility than theirs and is more robust against connectivity attacks (especially simplification). Robustness against geometry attacks is quite similar: our algorithm is globally more robust to strong distortions while their method is better against small ones. One exception is the smoothing attack which introduces obvious shrinkage deformations on this relatively sparse surface and thus makes our method fail. In general, their methods have difficulties under strong non-uniform simplification since the calculated mesh center can be wrongly moved towards the mesh part where the vertex density is higher, as mentioned in Section II-A. In all, our method is particularly suitable for the protection of dense meshes, for which the imperceptibility and the robustness against simplification are the main concerns. The advantage of their algorithms is that the watermark can resist attacks that introduce much higher objective distortions than its embedding. Neither method achieves the robustness against strong local deformation and cropping. In order to resist these attacks combined with connectivity changes, we may need to devise a robust and blind characteristic point extraction algorithm or a robust and blind mesh segmentation algorithm, which are quite difficult tasks. Rondao-Alface et al. [26] have done some work in this direction; however, the robustness of their method seems still to be improved.

Concerning the watermark capacity, the methods of Cho et al. can keep a constant capacity of 64 bits while the capacity of our scheme depends on the mesh shape (varying from 45 bits to 75 bits). In the future, we would like to exploit the possibility of ensuring a minimum capacity while keeping the other performances. Finally, our method outperforms their algorithms in terms of security. In the latter, no secret key is used and the modified histogram is exposed to everyone. The intrinsic easy accessibility of the global vertex norm histogram makes the security improvement difficult. On the contrary, a secret key is used in our algorithm for the SCS moment quantization and the current parameter setting ensures a relatively good secrecy of this key.

TABLE X
ROBUSTNESS EVALUATION OF THE WATERMARKED HORSE BY ALGORITHM I OF CHO ET AL. ($\alpha = 0.03$, 46 BITS INSERTED)

Attack	BER	Corr.	Our Corr.
0.10% noise	0	1	0.98
0.30% noise	0.24	0.52	0.86
0.50% noise	0.41	0.17	0.77
10-itera. smoothing	0	1	1
50-itera. smoothing	0.09	0.84	0.87
100-itera. smoothing	0.20	0.62	0.74
9-bit quantization	0.17	0.66	1
8-bit quantization	0.37	0.26	0.70
7-bit quantization	0.46	0.08	0.49
90% simplification	0.22	0.58	1
95% simplification	0.22	0.57	0.96
97.5% simplification	0.30	0.40	0.87
50% non-unif. simplifi.	0.11	0.80	0.83
75% non-unif. simplifi.	0.22	0.56	0.78
100% uniform remeshing	0	1	1
50% uniform remeshing	0.24	0.52	0.91

TABLE XI
ROBUSTNESS EVALUATION OF THE WATERMARKED BUNNY BY ALGORITHM II OF CHO ET AL. ($\alpha = 0.07$, 64 BITS INSERTED)

Attack	BER	Corr.	Our Corr.
0.10% noise	0	1	0.98
0.30% noise	0	1	0.85
0.50% noise	0.17	0.69	0.77
10-itera. smoothing	0.03	0.94	0.73
30-itera. smoothing	0.16	0.69	0.62
50-itera. smoothing	0.22	0.57	0.27
9-bit quantization	0.02	0.97	0.91
8-bit quantization	0.06	0.88	0.91
7-bit quantization	0.47	0.07	0.70
70% simplification	0.09	0.81	1
90% simplification	0.34	0.32	0.73
95% simplification	0.55	-0.09	0.74
25% non-unif. simplifi.	0.07	0.87	1
50% non-unif. simplifi.	0.48	0.03	0.73
midpoint subdivision	0.02	0.97	1
m-butterfly subdivision	0.02	0.97	1
Loop subdivision	0.09	0.81	0.71
100% uniform remeshing	0.02	0.97	0.94
50% uniform remeshing	0.22	0.57	0.74

VII. CONCLUSION AND FUTURE WORK

In this paper, a new robust and blind polygonal mesh watermarking algorithm is proposed. The watermark bits are inserted by slightly deforming the selected cover patches obtained after a simple mesh decomposition in the cylindric coordinate system. Watermark imperceptibility is ensured by using a smooth low-frequency mask to modulate the patch deformation; besides, the causality problem is resolved by introducing a compensation post-processing step. The robustness of this approach is due to the stability of the global and local (in patches) volume moment values under geometry, connectivity and even representation conversion attacks as long as they do not seriously modify the intrinsic shape (i.e. visual appearance) of the mesh. Finally, the security is explicitly considered by using a modified key-dependent informed quantization scheme with an appropriate parameter setting.

The proposed method can be improved in several aspects. First, it should be promising to introduce a perceptual distance metric to drive the patch deformation. An adaptable and robust mesh decomposition that produces patches with similar sizes is also of interest since it may allow to insert more bits without degrading imperceptibility and robustness; this "intelligent" decomposition may also be helpful to resolve the desynchronization problem caused by the patch classification. For long terms, we plan to investigate solutions to achieving robustness against

cropping/strong local deformation, combined with connectivity changes; this may resort to the design of a locally robust mesh shape descriptor.

ACKNOWLEDGMENTS

The authors would like to thank Dr. D. Coeurjolly for his help on voxelization and Marching Cubes algorithms.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. Morgan Kaufmann Publishers Inc., 2001.
[2] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications*. Marcel Dekker Inc., 2004.
[3] 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, 2008, (to appear).
[4] A. G. Bors, "Watermarking mesh-based representations of 3-D objects using local moments," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 687–701, 2006.
[5] O. Sorkine, D. Cohen-Or, and S. Toledo, "High-pass quantization for mesh encoding," in *Proc. of the Eurographics/ACM Siggraph Symposium on Geometry Processing*, 2003, pp. 42–51.
[6] F. Cayre, C. Fontaine, and T. Furon, "Watermarking security: Theory and practice," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, pp. 3976–3987, 2005.
[7] R. Ohbuchi, H. Masuda, and M. Aono, "Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models." *Computer Communications*, vol. 21, no. 15, pp. 1344–1354, 1998.
[8] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 939–949, 2003.
[9] S. Kanai, H. Date, and T. Kishinami, "Digital watermarking for 3D polygons using multiresolution wavelet decomposition," in *Proc. of the International Workshop on Geometric Modeling: Fundamentals and Applications*, 1998, pp. 296–307.
[10] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. of the ACM Siggraph*, 1999, pp. 49–56.
[11] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A frequency-domain approach to watermarking 3D shapes," *Computer Graphics Forum*, vol. 21, no. 3, pp. 373–382, 2002.
[12] J. Wu and L. Kobbelt, "Efficient spectral watermarking of large meshes with orthogonal basis functions," *The Visual Computer*, vol. 21, no. 8-10, pp. 848–857, 2005.
[13] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maître, "Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry," *Signal Processing: Image Communications*, vol. 18, no. 4, pp. 309–319, 2003.
[14] F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3D models," in *Proc. of the ACM Multimedia and Security Workshop*, 2004, pp. 143–154.
[15] P. Rondao-Alface and B. Macq, "Blind watermarking of 3D meshes using robust feature points detection," in *Proc. of the IEEE International Conference on Image Processing*, 2005, pp. 693–696.
[16] K. Wang, G. Lavoué, F. Denis, and A. Baskurt, "Hierarchical watermarking of semi-regular meshes based on wavelet transform," *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 4, 2008, (to appear).
[17] O. Benedens, "Geometry-based watermarking of 3D models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46–55, 1999.
[18] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Trans. on Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596–607, 2005.
[19] J.-W. Cho, R. Prost, and H.-Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. on Signal Processing*, vol. 55, no. 1, pp. 142–155, 2007.
[20] C. Zhang and T. Chen, "Efficient feature extraction for 2D/3D objects in mesh representation," in *Proc. of the IEEE International Conference on Image Processing*, 2001, pp. 935–938.
[21] A. Tuzikov, S. Sheynin, and P. V. Vasiliev, "Computation of volume and surface body moments," *Pattern Recognition*, vol. 36, no. 11, pp. 2521–2529, 2003.
[22] H. S. Kim and H. K. Lee, "Invariant image watermark using Zernike moments," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 766–775, 2003.
[23] M. Alghoniemy and A. H. Tewfik, "Geometric invariance in image watermarking," *IEEE Trans. on Image Processing*, vol. 13, no. 2, pp. 145–153, 2004.
[24] Y. Xin, S. Liao, and M. Pawlak, "Circularly orthogonal moments for geometrically robust image watermarking," *Pattern Recognition*, vol. 40, no. 12, pp. 3740–3752, 2007.
[25] A. Kalivas, A. Tefas, and I. Pitas, "Watermarking of 3D models using principal component analysis," in *Proc. of the IEEE International Conference on Multimedia & Expo*, 2003, pp. 637–640.
[26] P. Rondao-Alface, B. Macq, and F. Cayre, "Blind and robust watermarking of 3D models: How to withstand the cropping attack?" in *Proc. of the IEEE International Conference on Image Processing*, 2007, pp. V465–V468.
[27] M. S. Floater and K. Hormann, *Advances in Multiresolution for Geometric Modelling*. Springer-Verlag, 2005, ch. Surface parameterization: A tutorial and survey, pp. 157–186.
[28] J. J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, 2003.
[29] F. Pérez-González, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: A high-rate data-hiding method invariant to gain attacks," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, pp. 3960–3975, 2005.

[30] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.
[31] L. Pérez-Freire, P. Comesaña, and F. Pérez-González, "Information-theoretic analysis of security in side-informed data hiding," in *Proc. of the International Workshop on Information Hiding*, 2005, pp. 131–145.
[32] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.
[33] 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-IS & T Electronic Imaging*, vol. 6312, 2006, p. 63120L.
[34] G. Taubin, "Geometric signal processing on polygonal meshes," in *Proc. of the Eurographics State-of-the-art Reports*, 2000, pp. 81–96.
[35] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. of the ACM Siggraph*, 1997, pp. 209–216.
[36] D. Zorin and P. Schröder, "Subdivision for modeling and animation," in *Proc. of the ACM Siggraph Course Notes*, 2000.
[37] M. Attene and B. Falcidieno, "ReMESH: An interactive environment to edit and repair triangle meshes," in *Proc. of the IEEE International Conference on Shape Modeling and Applications*, 2006, pp. 271–276.
[38] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Proc. of the ACM Siggraph*, 1987, pp. 163–170.