

Technical Section

Robust and blind mesh watermarking based on volume moments[☆]Kai Wang^{a,*}, Guillaume Lavoué^a, Florence Denis^b, Atilla Baskurt^a^a Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France^b Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France

ARTICLE INFO

Article history:

Received 7 April 2010

Received in revised form

10 September 2010

Accepted 15 September 2010

Available online 22 September 2010

Keywords:

3D mesh

Digital watermarking

Robustness

Blindness

Volume moment

Imperceptibility

Causality problem

ABSTRACT

This paper presents a robust and blind watermarking algorithm for three-dimensional (3D) meshes. The watermarking primitive is an intrinsic 3D shape descriptor: the analytic and continuous geometric volume moment. During watermark embedding, the input mesh is first normalized to a canonical and robust spatial pose by using its global volume moments. Then, the normalized mesh is decomposed into patches and the watermark is embedded through a modified scalar Costa quantization of the zero-order volume moments of some selected candidate patches. Experimental results and comparisons with the state of the art demonstrate the effectiveness of the proposed approach.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Recent advances in 3D acquisition technologies, 3D graphics rendering and geometric modeling have boosted the creation of 3D model archives in many applications such as medical imaging, scientific simulation, cultural heritage, digital entertainment and computer-aided design. 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 and the intranets. Unfortunately, like digital images and audio/video clips, 3D graphic models can be easily duplicated and redistributed without any loss of quality by a pirate. This illegal behavior infringes the copyright of graphic model owners and can also do harm to the whole underlying commercial chain. Therefore, under this background, there now exists a critical demand on the intellectual property protection of 3D models, mostly represented as polygonal meshes [1]. Digital watermarking [2,3] is considered as an efficient solution to solve this emerging problem.

The basic idea of the digital watermarking technique is to hide a piece of secret information, i.e. the watermark, within the functional part of a multimedia content (often called *cover*

content in watermarking terminology). According to the aimed application, we distinguish between the *robust* watermark used for intellectual property protection and the *fragile* watermark used for content authentication. A robust watermark has to be as resistant as possible against various attacks on the watermarked content, while keeping itself imperceptible. A fragile watermark is intentionally designed to be vulnerable to certain non-tolerable operations, and its extraction failure indicates the existence of such operations on the watermarked content. According to whether the original cover content is required or not at the watermark extraction stage, watermarking algorithms can also be classified as *non-blind* schemes or *blind* schemes. Blind schemes are preferred in real-world applications, since in many cases the original content cannot or even should not be present at the extraction stage, often due to efficiency and security issues. For example, in the copy control examination application, it is inappropriate to make the original copy available in the control device that is probably in the hand of a malicious client.

This paper focuses on the robust and blind watermarking of 3D meshes. Our main objective is to achieve a better robustness against the intractable *connectivity attacks* for blind mesh watermarks. Different from *geometry attacks* that only modify the coordinates of the mesh vertices (e.g. noise addition, smoothing and vertex coordinate quantization), connectivity attacks (e.g. surface simplification and remeshing) can completely change the positions and adjacency relationships of the mesh vertices while well preserving the global shape of the watermarked model. According to the surveys of Rondao-Alface and Macq [4] and of Wang et al. [5], connectivity attacks can actually defeat most

[☆]This article was recommended for publication by R. Pajarola.

* Corresponding author.

E-mail addresses: kwang@liris.cnrs.fr (K. Wang), glaouve@liris.cnrs.fr (G. Lavoué), fdenis@liris.cnrs.fr (F. Denis), abaskurt@liris.cnrs.fr (A. Baskurt).

of the existing blind mesh watermarking schemes. Another difficult attack is the *3D object representation conversion* (e.g. from mesh to voxels): the mesh model itself no longer exists after such a conversion. All the aforementioned attacks are rather common operations in many mesh applications and may also be conducted by a pirate who attempts to remove the watermark. When performing these operations, normally the user also tries to preserve the basic shape (i.e. the visual appearance) of the model. Indeed, a too much distorted object does not present too much interest to the user and the application. Following this idea, we believe that a valuable robust mesh watermarking method has to be linked to the basic *3D shape* that is behind the mesh, but not to the mesh itself. Hence, we have chosen an intrinsic 3D shape descriptor, i.e. the *geometric volume moment*, as the watermarking primitive in which we embed the watermark bits. This descriptor is of continuous nature and depends only on the 3D analytic shape represented by the mesh; therefore, it should be robust against geometry, connectivity and representation conversion attacks providing that they do not seriously modify the shape of the watermarked model. In our method, a robust and blind watermark is embedded in the cover mesh by slightly modifying these geometric moments through a quantization-based technique.

Another critical issue for blind mesh watermarking is the so-called *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 [6], the author first establishes an order for the watermarking candidate vertices according to a geometric criterion, and then modifies another correlated geometric quantity to insert watermark bits. The original vertex order may be altered after the bit insertion, and it is necessary to introduce a post-processing step so as to recover this order. In our method presented later in this paper, a geometric *compensation* process is performed after watermark embedding, which recovers some important features of the cover mesh and thus clearly resolves the causality problem encountered by our algorithm. Finally, the watermark *imperceptibility* has also to be carefully taken into account. It has been shown that watermark embedding in the mesh low-frequency components can be both more robust and more imperceptible [7,8]. We have followed this principle when devising our method.

Hence, we present here a new robust and blind mesh watermarking algorithm that is based on the mesh's continuous and analytic volume moments. The most important feature of the proposed method is its strong robustness against the intractable connectivity attacks. The remainder of this paper is organized as follows: Section 2 briefly reviews the related work; Section 3 provides an overview of the proposed method; Sections 4 and 5 detail the watermark embedding and extraction procedures; Section 6 presents some experimental results, including comparisons with some state-of-the-art schemes obtained by using a recently developed benchmarking system; finally, Section 7 concludes the paper and proposes several future working directions.

2. Related work

Relatively few robust and blind watermarking algorithms have been proposed for 3D meshes. As pointed out in [5], this situation is mainly due to two difficulties: (1) the intrinsic irregular sampling nature of 3D meshes, and (2) the existence of a large number of intractable attacks on the watermarked model.

In the literature of 3D mesh watermarking, the blindness has been achieved in several *spatial-domain-based* algorithms [6,9–12]. These schemes has a certain level of robustness against common geometry attacks and even cropping, but are in general

fragile to connectivity attacks because the used geometric watermarking primitives may disappear or be seriously disturbed after such attacks. On the contrary, some *transform-domain-based* algorithms [13–16] are robust but non-blind. The used transformation tools are sensitive to connectivity changes; hence, a resampling preprocessing step is needed at extraction, so as to recover the original connectivity configuration or to fix the cut part after a cropping attack. This step ensures a sufficient level of robustness but unavoidably makes the scheme non-blind. There exist several blind algorithms in transformed domains [17–21]; however, they are not that robust against connectivity attacks and some of them also have applicability restrictions.

Several blind and robust algorithms have been nevertheless proposed. In order to achieve the robustness to connectivity attacks, these methods use statistical mesh shape descriptors as watermarking primitives, such as the average normal direction of the facets in a patch [22], the histogram of the vertex coordinate prediction errors [23], or the histogram of the vertex norms [24]. These algorithms are either blind [23,24] or semi-blind [22] and achieve relatively good robustness due to the intrinsic stability of the shape descriptor primitives. The methods of Cho et al. [24] may have been the most robust blind algorithms proposed so far. The authors first construct the histogram of the distances between vertices and mesh center, and then divide this histogram in bins associated with different ranges of the vertex-center distance. Afterwards, they make the hypothesis of a uniform histogram distribution in the obtained bins. One bit is embedded in each bin by slightly modifying the mean value (or the variance) of the distribution in the bin. In despite of the strong robustness against most of the common attacks, it seems that their schemes have two drawbacks: first, they tend to introduce visible ring-like distortions on watermarked models; second, the mesh center is simply calculated as the average of all the vertices in the model, which is not very stable under spatially non-uniform (i.e. anisotropic) simplification and resampling attacks.

3. Overview of the proposed method

As mentioned in Section 1, we make the assumption that a good mesh watermarking primitive has to be intrinsically linked to the 3D shape represented by the mesh. In this sense, the analytic and continuous volume moment [25–27] seems promising to become an effective primitive. The local (resp. global) mesh volume moment of a certain order (p,q,r) is actually the (continuous) volume integration of the function $f(x,y,z) = x^p y^q z^r$ within part (resp. entire) of the mesh model, and its value is very stable under various common attacks (details and proofs will be presented in the subsequent sections). Our objective here is to embed in the moments of a 3D mesh a sequence of bits, i.e. a *multi-bit watermark*. However, two difficulties arise: first, for a given mesh, its global moments of different orders are correlated, therefore it is complicated to modulate these moments simultaneously and independently (so as to embed in them the individual bits of the watermark sequence); second, the volume moment transformation is not reversible, thus it is difficult to modify the moment to a certain target value in an easy and straightforward way. The first point forced us to decompose the mesh into several patches and insert one bit into each patch. For the second point, an efficient iterative patch deformation algorithm has been devised. This algorithm ensures that the patches rapidly reach their wanted moment values while not introducing noticeable distortions.

Fig. 1(a) illustrates the bloc diagram of our watermark embedding procedure. The cover mesh is first normalized to a canonical pose by using its global volume moments. Then, the

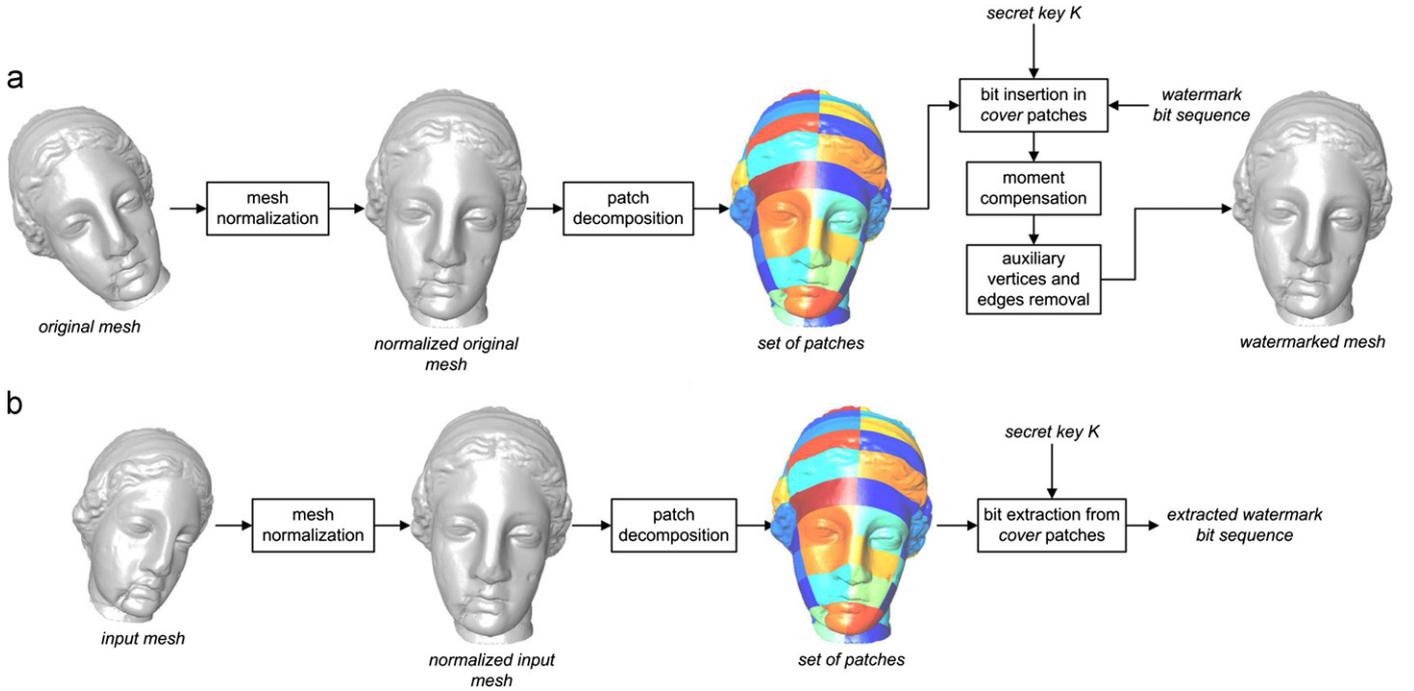


Fig. 1. Block diagrams of (a) the watermark embedding and (b) the watermark extraction procedures.

mesh vertex coordinates are transformed from the Cartesian system (x,y,z) to the cylindrical system (h,r,θ) . Afterwards, the normalized mesh is decomposed into patches by discretizing its h and θ domains in the cylindrical system. For several selected patches (the so-called *cover patches*), we calculate their zero-order volume moments and quantize them so as to embed one bit per patch. The moment quantization scheme is a modified version of the widely used scalar Costa scheme (SCS) [28]. Similar to the original SCS, a secret key K is used in our modified scheme to construct the pseudo-random quantization codebooks for the patch moments. Note that in order to ensure a precise patch moment calculation, we need to insert some auxiliary vertices and edges on the patch borders; these auxiliary elements can be easily removed after the watermark embedding. The patch moment modification is realized by using an iterative deformation process. The induced deformation is also modulated by a smooth mask so as to keep it imperceptible to the human visual system. The third difficulty, namely the causality problem, occurs at this stage, since 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 is introduced to resolve this problem.

Fig. 1(b) illustrates the block diagram of our blind watermark extraction procedure that consists of three main steps: mesh normalization, patch decomposition, and bit extraction from the cover patches. Sections 4 and 5 will elaborate on the technical details of the proposed watermark embedding and extraction algorithms.

4. Watermark embedding

4.1. Preliminaries on volume moments

Before presenting the different steps of the watermark embedding procedure, it would be necessary to briefly introduce the concept and calculation of the volume moment. For a closed 3D surface \mathcal{S} , its geometric volume moments (of different orders)

are defined as follows:

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

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 \mathcal{S} ; otherwise it is equal to 0). The volume moment of order p,q,r is actually the (continuous) volume integration of the function $f(x,y,z) = x^p y^q z^r$ inside the closed surface \mathcal{S} . For an orientable 3D mesh, Zhang and Chen [25] and Tuzikov et al. [26,27] derived independently the explicit expression for this integration. The basic idea is to calculate it as a sum of signed integrations over several elementary volumes. For a triangular mesh, the elementary volume 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 according to the orientation of f_i and the relative position between f_i and \mathcal{O} . Note that if the facets are correctly oriented (i.e. the normals of the facets all point to the outside of the closed surface), then the zero-order 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 in the following as Eqs. (2)–(5), where $f_i = \{v_{i1}, v_{i2}, v_{i3}\} = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}), (x_{i3}, y_{i3}, z_{i3})\}$. A more complete list of the elementary moment calculation expressions can be found in the papers of Tuzikov et al. [26,27]:

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

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

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

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

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 coordinate system origin, and then calculating the moments of the obtained closed surface. Volume moments are very robust geometric features and have been used in mesh self-registration and 3D shape retrieval [25]. In the proposed method, we will use the global volume moments for mesh normalization and the local volume moments as watermarking primitives.

4.2. Mesh normalization

Mesh normalization is used as a preprocessing step by both the watermark embedding and the watermark extraction algorithms, and consists of 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; and
3. rotation of the mesh so that its three principal axes coincide with the axes of the Cartesian coordinate system.

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

$$C = (x_c, y_c, z_c) = \left(\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}} \right). \quad (6)$$

The principal axes of the mesh are obtained as the ordered eigenvectors (according to their associated eigenvalues) of the following matrix:

$$M = \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix}. \quad (7)$$

In our implementation, the most significant principal axis is aligned with the axis Z, the second significant axis with the axis Y, and the least significant axis with the axis X. In order to resolve the axis alignment ambiguity problem, besides the compliance to the right-hand rule of the three principle axes, we need to impose other geometric constraints (e.g. the global moments m_{300} and m_{030} of the rotated mesh should be positive, as proposed in [25]). In this way, we ensure that the obtained aligned object is unique and consistent. Note that the volume moments m_{100} , m_{010} , m_{001} , m_{110} , m_{101} and m_{011} of the normalized mesh are all equal to zero.

The above normalization relies on the analytic volume moments and therefore is processed in a continuous space. So far, in most existing watermarking methods, the mesh normalization step depends entirely on the vertex coordinates, while completely discarding the mesh connectivity information [23,24,29]. This kind of “discrete” moment is not very robust, especially against anisotropic connectivity attacks. Recently, Rondao-Alface et al. [30] have calculated the mesh center as the

weighted average position of the mesh vertices, which is somewhat equivalent to the calculation based on the mesh *surface* moments [27]. Table 1 compares the robustness of the mesh normalizations based on discrete, surface and volume moments, in terms of the center norm variation $V_{|C|}$ and the maximum principal axis variation MV_{PA} (in degree). The experiments were carried out on the Venus mesh (100 759 vertices) that is illustrated in Fig. 1. The conducted attacks in the experiments are deemed to have very strong amplitudes in the context of mesh watermarking (cf. Fig. 9, the visual effects of such attacks can be easily perceived). From Table 1, it can be observed that the mesh normalization based on volume moments has the best overall performance, especially under anisotropic noise addition and simplification.

4.3. Decomposing the mesh into patches

The mesh is then decomposed into patches so as to insert one bit per patch. After the above normalization step, each vertex $v_k = (x_k, y_k, z_k)$ is converted into cylindrical coordinate system as $v_k = (h_k, r_k, \theta_k) = (z_k, \sqrt{x_k^2 + y_k^2}, \tan^{-1}(y_k/x_k))$. The patch decomposition is carried out in the obtained cylindrical system, and is simply a uniform discretization of the h and θ domains of the mesh object into I_h and I_θ intervals with two steps h_{step} and θ_{step} .

Each vertex is associated to its proper patch by calculating its discretized indices $ind(h_k) \in \{0, 1, \dots, I_h - 1\}$ and $ind(\theta_k) \in \{0, 1, \dots, I_\theta - 1\}$; however, some facets may cover several different patches. These facets have to be split into a number of 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 to the watermark robustness. The task is accomplished by automatically adding auxiliary vertices and edges on the patch borders (cf. Fig. 2). 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 that are denoted by $\mathcal{P}_j, j \in \{0, 1, \dots, I_h \cdot I_\theta - 1\}$. These patches are ordered according to their spatial locations, and the patch index is determined as $j = ind(h_k) \cdot I_\theta + ind(\theta_k)$, where $ind(h_k)$ and $ind(\theta_k)$ are the discretized indices of any belonging vertex in the interior of the patch.

I_h and I_θ are two important parameters of our algorithm: if we increase the patch number, the watermark *payload* (i.e. the number of embedded bits) 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 patch contains just one vertex), the final deformation will become of high frequency, which is more visible and less robust. The setting of $I_h = 11$ and $I_\theta = 8$ seems to achieve a good trade-off between watermark payload, robustness and imperceptibility for most meshes. An adaptable setting of these two parameters according to the individual mesh shape constitutes one part of our future work.

Table 1

Robustness comparison of the different mesh normalizations on the Venus model under various strong-amplitude attacks.

Attack	Discrete moments		Surface moments		Volume moments	
	$V_{ C }$	MV_{PA} (deg)	$V_{ C }$	MV_{PA} (deg)	$V_{ C }$	MV_{PA} (deg)
0.50% noise	1.1×10^{-6}	0.003	6.4×10^{-4}	0.23	3.7×10^{-5}	0.01
7-bit quantization	1.6×10^{-5}	0.01	3.0×10^{-3}	1.07	2.7×10^{-5}	0.05
90% simplification	3.3×10^{-3}	3.32	4.0×10^{-4}	0.05	1.2×10^{-4}	0.03
0.50% anisotropic noise	8.0×10^{-6}	0.01	4.4×10^{-2}	5.70	2.4×10^{-5}	0.01
50% anisotropic simplification	4.0×10^{-1}	82.53	2.3×10^{-3}	0.18	5.5×10^{-4}	0.05

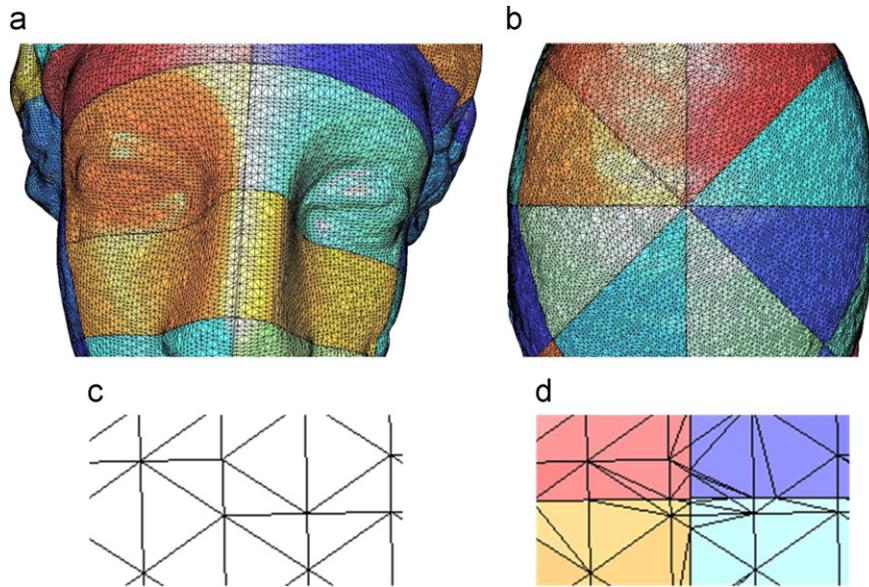


Fig. 2. (a), (b) and (d) illustrate three close-ups of a Venus head mesh that is decomposed into $I_h \times I_\theta = 11 \times 8 = 88$ patches; the original connectivity of (d) is shown in (c).

The combination “mesh normalization + cylindrical discretization” constitutes a simple but effective mesh decomposition process. 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 very robust. In order to verify this robustness, we have analyzed the stability of the zero-order volume moments of the generated patches under various attacks, including those that are spatially non-uniform. Fig. 3 presents the results on the Horse model (112 642 vertices) that is illustrated in Fig. 6(b). It can be seen that the patch moment values are highly stable under the tested strong-amplitude attacks, which also demonstrates the potential interest of using these local moments as watermarking primitives. One limitation of the proposed patching method is that it is not robust against strong local deformation and cropping, which are actually very difficult to handle for blind mesh watermarking schemes. These attacks cause serious de-synchronization problems (i.e. the generated patches in which watermark bits are embedded are not consistent) due to the severe deviation of the mesh normalization result.

4.4. Patch classification and watermark synchronization

The obtained patches are classified into three groups:

1. *cover* patches used for watermark bit embedding;
2. *discarded* patches that are not suitable to be deformed; and
3. *compensation* patches used for global moment compensation after watermark bit embedding.

The discarded patches will not be used for watermark bit embedding nor for moment compensation. They are actually some small patches with either a very low zero-order volume moment amplitude, or a very small h or θ domain range. It is in practice very difficult to deform these singular patches equally strongly as the other patches, and their volume moments are not that robust compared to the other ones; therefore they are discarded and will not be deformed in our watermark embedding algorithm. Three empirical thresholds $\bar{m}_{000} = 0.0005$ for zero-order volume moment amplitude, $\bar{h}_r = 0.35 \times h_{step}$ for h domain range, and $\bar{\theta}_r = 0.35 \times \theta_{step}$ for θ domain range are established to filter out these patches.

The compensation patches serve to be deformed after the watermark bit embedding in the cover patches, with the objective to recover the mesh center position and principal axis orientations. This mesh canonical pose recovery is necessary to prevent the causality problem and thus is critical to the correctness of the watermark embedding. 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 the watermark bit embedding and considered as compensation patches. They are hereafter denoted by $\mathcal{P}_l^c, l \in \{0, 1, \dots, 11\}$. 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 $\mathcal{P}_n^w, n \in \{0, 1, \dots, N-1\}$. A cover patch with a smaller index in this sequence also has a smaller index in the global indexing $\mathcal{P}_j, j \in \{0, 1, \dots, I_h \cdot I_\theta - 1\}$. This cover patch order is used for the watermark synchronization: watermark bits are sequentially embedded in or extracted from these ordered cover patches.

The above patch classification may induce watermark de-synchronization problem. For instance, after the watermark embedding or an attack, a compensation patch may become a cover patch if its m_{000} amplitude decreases. In order to prevent this problem, we take out some special measures for the potentially sensitive patches with regard to the patch classification. For example, the m_{000} amplitude of the compensation patch \mathcal{P}_{11}^c is constrained to be increased during the moment compensation. These measures preserve and enhance the original patch classification, and in consequence can effectively prevent the watermark de-synchronization from happening, even under strong attacks. In order to further reinforce the system's robustness against this problem, at the extraction side the algorithm is designed to realize several different bit extractions (normally less than 4 even under strong attacks) by classifying the suspicious patch(es) into different possible groups. The de-synchronization problem would be resolved by transmitting an additional sequence of $I_h \times I_\theta$ bits to the extraction side so as to explicitly indicate the locations of the cover patches; however, strictly speaking, this solution would make the algorithm semi-blind.

4.5. Patch moment quantization

After patch decomposition and classification, the next step is the watermark bit embedding. We have chosen the cover patches'

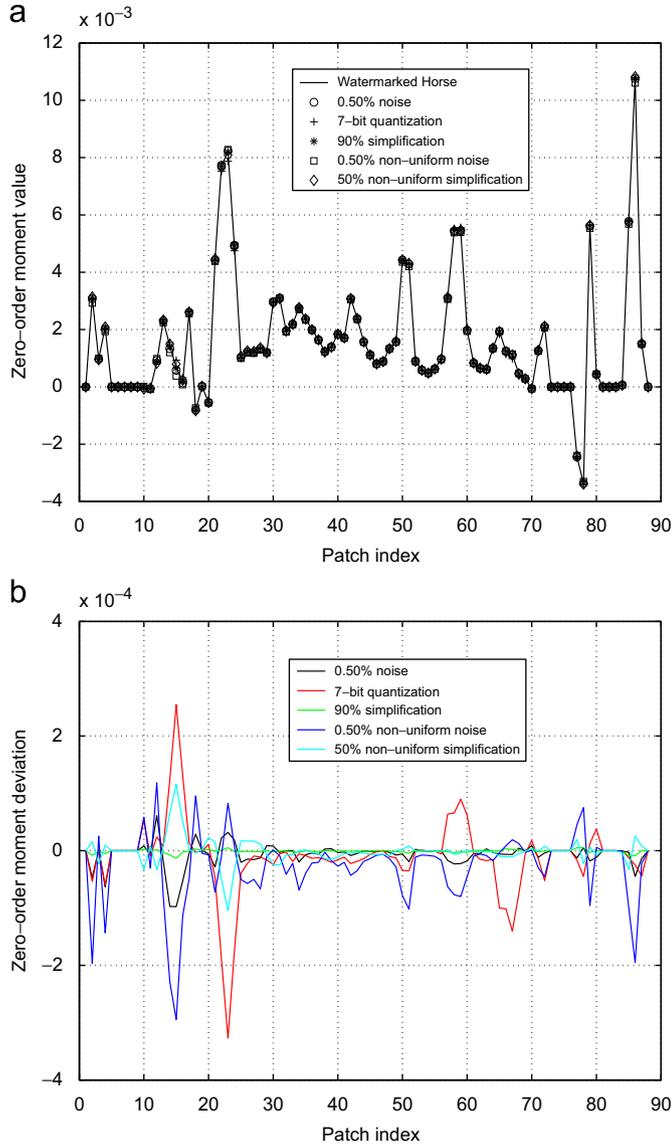


Fig. 3. Stability of the patches' zero-order moment values of the watermarked Horse model under various strong-amplitude attacks: (a) the different patch moment curves almost coincide and (b) the corresponding patch moment deviations are very small.

zero-order volume moments as the watermarking primitives, which are actually very robust local shape features (cf. Fig. 3). The n -th watermark bit $w_n \in \{0,1\}$ is inserted by quantizing the zero-order moment of the n -th cover patch that is denoted by $m_{000}^{(\mathcal{P}_n^w)}$. The proposed quantization scheme is a modified version of the conventional scalar Costa scheme (SCS) [28], which is a practical and suboptimal implementation of the ideal Costa scheme [31] and has been widely used in image, audio and video watermarking due to its easy implementation and its high flexibility between different metrics such as robustness, imperceptibility and payload.

In the following, we will present the details of our moment quantization method. First, for each $m_{000}^{(\mathcal{P}_n^w)}$, $n \in \{1, 2, \dots, N-1\}$ that is to be watermarked, a structured pseudo-random codebook is established as follows:

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

where $\Delta^{(\mathcal{P}_n^w)}$ is the quantization step, $z \in \mathcal{Z}$ is an integer, $a \in \{0,1\}$ is the watermark bit from the codeword u , and $t^{(\mathcal{P}_n^w)} \Delta^{(\mathcal{P}_n^w)}$ is an

additive pseudo-random dither signal. In our implementation, the pseudo-random numbers $t^{(\mathcal{P}_n^w)}$, $n \in \{1, 2, \dots, N-1\}$ constitute a simulation sequence of a random variable uniformly distributed in $[-\frac{1}{2}, \frac{1}{2}]$, and they are generated by using a secret key K . Note that the codewords in $\mathcal{U}_{m_{000}^{(\mathcal{P}_n^w)}, t^{(\mathcal{P}_n^w)}}$ represent bits of 0 and 1 in a uniform and interleaved manner.

Unlike in the conventional SCS, the quantization step $\Delta^{(\mathcal{P}_n^w)}$ in our method is no longer fixed for all the watermarking primitives, but is variant for the moment of each individual cover patch. A fixed step has been experimentally proven inappropriate for the patch moment quantization. Indeed, different patches can tolerate very different moment variations with respect to the watermark imperceptibility; meanwhile, a same attack also induces quite different moment variations on the patches of different sizes. Therefore, in order to ensure a roughly comparable watermarking performance (mainly in terms of imperceptibility and robustness) for each cover patch, it is better to use variant and “adaptive” moment quantization steps for individual patches. We propose the following derivation of the component-wise steps $\Delta^{(\mathcal{P}_n^w)}$, $n \in \{1, 2, \dots, N-1\}$:

$$\Delta^{(\mathcal{P}_n^w)} = \begin{cases} \Delta_{pre} \cdot \left| m_{000}^{(\mathcal{P}_{n-1}^w)} / \left[\frac{m_{000}^{(\mathcal{P}_{n-1}^w)}}{m_{000}^{(\mathcal{P}_n^w)}} \right] \right| & \text{if } \frac{m_{000}^{(\mathcal{P}_{n-1}^w)}}{m_{000}^{(\mathcal{P}_n^w)}} \geq 1, \\ \Delta_{pre} \cdot \left| m_{000}^{(\mathcal{P}_{n-1}^w)} \cdot \left[\frac{m_{000}^{(\mathcal{P}_n^w)}}{m_{000}^{(\mathcal{P}_{n-1}^w)}} \right] \right| & \text{if } \frac{m_{000}^{(\mathcal{P}_{n-1}^w)}}{m_{000}^{(\mathcal{P}_n^w)}} < 1, \end{cases} \quad (9)$$

where $m_{000}^{(\mathcal{P}_{n-1}^w)}$ is the watermarked moment value of the patch \mathcal{P}_{n-1}^w with $m_{000}^{(\mathcal{P}_0^w)} = m_{000}^{(\mathcal{P}_0^c)}$ (hence, the first cover patch \mathcal{P}_0^w is utilized as an anchor patch and is not watermarked; accordingly, a total number of $(N-1)$ bits can be embedded in the N cover patches of the mesh); and Δ_{pre} is given by

$$\Delta_{pre} = \begin{cases} 0.04 & \text{if } |m_{000}^{(\mathcal{P}_n^w)}| > 0.01, \\ 0.07 & \text{if } \bar{m}_{000} < |m_{000}^{(\mathcal{P}_n^w)}| \leq 0.01. \end{cases} \quad (10)$$

It can be inferred that the quantization step $\Delta^{(\mathcal{P}_n^w)}$ calculated by using Eq. (9) is approximately proportional to the patch moment amplitude $|m_{000}^{(\mathcal{P}_n^w)}|$. In consequence, normally, the patches with larger moment amplitudes can adaptively have larger moment variations. This is quite reasonable because we can easily prove that a same additive vertex coordinates modification induces larger moment variation on a patch with a larger moment amplitude. There are different Δ_{pre} values for the patches with moderate moment amplitudes and those with large amplitudes (cf. Eq. (10)). This distinction helps to further balance the induced distortions on these different patches and is also theoretically reasonable (cf. Appendix A). Although a more sophisticated derivation of Δ_{pre} may be possible, the empirical setting as given by Eq. (10) has already worked well enough in practice for most meshes.

It is worthwhile to point out that in order to get an “adaptive” quantization step, we cannot directly set $\Delta^{(\mathcal{P}_n^w)}$ proportional to $|m_{000}^{(\mathcal{P}_n^w)}|$ (i.e. $\Delta^{(\mathcal{P}_n^w)} = c \cdot |m_{000}^{(\mathcal{P}_n^w)}|$ with c a constant), since after the bit insertion through patch moment modification, the quantization step will be changed and in consequence we may not be able to correctly extract the inserted bit even under no attack. In Eq. (9), an approximative proportionality between $\Delta^{(\mathcal{P}_n^w)}$ and $|m_{000}^{(\mathcal{P}_n^w)}|$ is achieved by introducing the term $m_{000}^{(\mathcal{P}_{n-1}^w)}$ (i.e. the quantized moment value of the previous patch \mathcal{P}_{n-1}^w) and the integer rounding operator ($\lceil \cdot \rceil$ or $\lfloor \cdot \rfloor$). The integer rounding operation is necessary since in most cases it ensures that the quantization step is kept unchanged before and after the bit insertion, as long as the

ceiled or floored integer ratio between $m_{000}^{(\hat{\mathcal{P}}_n^w)}$ and the watermarked moment value $m_{000}^{(\mathcal{P}_n^w)}$ (determined according to Eq. (9), with $m_{000}^{(\mathcal{P}_n^w)}$ in the place of $m_{000}^{(\hat{\mathcal{P}}_n^w)}$) is the same as that between $m_{000}^{(\mathcal{P}_{n-1}^w)}$ and the original moment value $m_{000}^{(\mathcal{P}_n^w)}$. However, it is still possible that the integer ratio changes after moment modification and therefore the quantization step also changes. When this situation is encountered (actually it rarely occurs), the patch moment is automatically re-quantized by a specific algorithm which attempts to correctly embed the watermark bit while not introducing too much patch deformation.

The idea of setting the quantization step of the patch \mathcal{P}_n^w related to the quantized moment of its previous patch \mathcal{P}_{n-1}^w was also partially inspired by the work of Pérez-González et al. [32]. Their rational dither modulation (RDH) method achieves the invariance to the value-metric scaling attacks for the quantization index modulation watermarking paradigm [33]. We have proposed the above RDH-like scheme in part to reinforce the watermark robustness against the alteration of the farthest vertex (from the mesh center) that is used by the uniform scaling operation during the mesh normalization step (cf. Section 4.2). This alteration is possible after watermark embedding or after attacks. It can be seen that with the derivation of the quantization step $\Delta^{(\mathcal{P}_n^w)}$ as given in Eq. (9), the proposed moment quantization scheme is intrinsically invariant to uniform scaling, since the codewords in $\mathcal{U}_{m_{000}^{(\mathcal{P}_n^w)}, t^{(\mathcal{P}_n^w)}}$ vary proportionally with the watermarking primitives (i.e. the patch moments) under this operation. In consequence, we can effectively enhance the watermark robustness against the mesh local scaling caused by the alteration of the farthest vertex.

The next step of the moment quantization is to find in the constructed codebook the nearest codeword $u_{m_{000}^{(\mathcal{P}_n^w)}}$ to $m_{000}^{(\mathcal{P}_n^w)}$, which also correctly embeds the watermark bit w_n . The latter point means that w_n should be equal to value a in the derivation of $u_{m_{000}^{(\mathcal{P}_n^w)}}$ as given in Eq. (8). The quantized value $m_{000}^{(\hat{\mathcal{P}}_n^w)}$ is then calculated as follows:

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

where $\alpha^{(\mathcal{P}_n^w)} \in [0, 1]$ is the so-called distortion compensation (DC) factor. We always select an appropriate value for $\alpha^{(\mathcal{P}_n^w)}$ so as to ensure the correctness of the watermark bit extraction when there is no attack. The above bit embedding procedure consists in pushing $m_{000}^{(\mathcal{P}_n^w)}$ towards $u_{m_{000}^{(\mathcal{P}_n^w)}}$, to within the interval $(u_{m_{000}^{(\mathcal{P}_n^w)}} - \Delta^{(\mathcal{P}_n^w)})/4, u_{m_{000}^{(\mathcal{P}_n^w)}} + \Delta^{(\mathcal{P}_n^w)}/4$, which is the decoding area of $u_{m_{000}^{(\mathcal{P}_n^w)}}$ under the nearest neighbor criterion.

4.6. Patch deformation

The next step is to deform the cover patches so as to reach their quantized moment values $m_{000}^{(\hat{\mathcal{P}}_n^w)}$, $n \in \{1, 2, \dots, N-1\}$. As it is not easy to deduce the new coordinates of the patch vertices directly from the quantized moment value, we need to modify the moment of a cover patch heuristically and iteratively by moving its comprised vertices. The amplitude and direction of this patch deformation is adjusted in each iteration step so that the patch's 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 mask function that is illustrated in Fig. 4, so that the patch's global deformation is of low frequency and thus invisible. Actually, as mentioned in Section 1, the human visual system is less sensitive to the modification of the mesh low-frequency components than that of the high-frequency components [7,8].

In each step of the iterative deformation process, every vertex has its own multiplicative deformation factor. For a vertex $v_k = (h_k, r_k, \theta_k)$ within \mathcal{P}_n^w , the derivation of its deformation factor s_{v_k} begins with a normalization of its cylindrical coordinates h_k and θ_k :

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

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

where h_k' and θ_k' are the normalized coordinates, $h_{\max}^{(\mathcal{P}_n^w)}$ and $h_{\min}^{(\mathcal{P}_n^w)}$ ($\theta_{\max}^{(\mathcal{P}_n^w)}$ and $\theta_{\min}^{(\mathcal{P}_n^w)}$) are, respectively, the maximum and the minimum h domain (θ domain) coordinates of all the vertices within \mathcal{P}_n^w or on the borders of \mathcal{P}_n^w . Under this normalization, the vertices close to the patch borders will have small h_k' and θ_k' values, while the vertices close to the patch center will receive large values. For each vertex, two weights are then calculated: the following equation gives the formula for the h domain weight calculation, the calculation of the θ domain weight $wt_{\theta_k'}$ has a similar form:

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

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

$$s_{v_k} = \begin{cases} 1 + wt_{h_k'} \cdot wt_{\theta_k'} & \text{if } s > 1, \\ 1 - wt_{h_k'} \cdot wt_{\theta_k'} & \text{if } s < 1. \end{cases} \quad (15)$$

At the end of each iteration step, the Cartesian coordinates of a candidate displaced vertex are obtained as the multiplication of its original coordinates (x_k, y_k, z_k) with s_{v_k} or $(2 - s_{v_k})$, depending on the moment contribution sign of its incident facets (cf. step 7 of Algorithm 1 for more details). The motivation of setting the weight function as in Eq. (14) is to obtain a relatively simple function which varies smoothly from 0 (near the patch boundaries) to a maximum value (near the patch center).

The deformation mask function indicates the relationship between the vertices' local deformation factors s_{v_k} and their cylindrical coordinates h_k and θ_k . As illustrated in Fig. 4, this mask function has a very smooth shape: the function value is constant in the border and center regions, and has a sinus-like shape between the above two regions. The amplitude and direction of this mask are dependent on the global deformation factor s and vary in each iteration step.

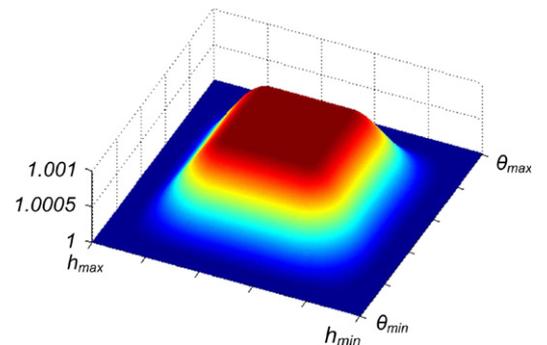


Fig. 4. An example of the deformation mask function. Here the global deformation factor s is equal to 1.001.

The objective now is to find, for each patch, the correct value for s that produces the target quantized moment function when applying the corresponding deformation mask function on the original patch. For this purpose, we have devised a simple and efficient iterative process, which is summarized as Algorithm 1. Note that some vertices are not modifiable during the patch deformation. These non-modifiable vertices include the added border vertices, the direct neighbors of the border vertices, and the vertices having simultaneously facets with positive and negative moment contributions. We have also constrained that a displaced vertex cannot get out of its original patch. By using this iterative algorithm, normally the target moment value can be attained within less than 25 iterations. Actually, each patch may have its own deformation mask function shape. For example, we can change the support and frequency of the sinus function in Eq. (14), or we can use a different smooth function (e.g. a Gaussian-like function) in the place of the sinus function. Experimentally, there is not too much difference in imperceptibility for these different functions, and a uniform setting of the above mask function shape for all the patches already ensures a satisfying performance of the proposed watermarking method.

Algorithm 1. Iterative patch deformation algorithm.

- Notations:** s is the global deformation factor; k_s is the modification step of s ; $m_{000}^{(P_n^w)}$ is the original moment of the patch; $\hat{m}_{000}^{(P_n^w)}$ is the target moment value; and m_i is the zero-order moment of the deformed patch after i -th iteration
- 1: Determine the comprised vertices of the current patch P_n^w ; for each comprised vertex deduce its modifiability; for each modifiable vertex v_k record its original Cartesian coordinates (x_k, y_k, z_k)
 - 2: Initialize the parameters: $s = 1$, $k_s = 0.01$, $i = 1$, $m_{-1} = m_0 = m_{000}^{(P_n^w)}$
 - 3: **repeat**
 - 4: Modify s according to the following rule
 - if $m_{i-1} < \hat{m}_{000}^{(P_n^w)}$ and $m_{i-2} < \hat{m}_{000}^{(P_n^w)}$, then $s \leftarrow s + k_s$;
 - if $m_{i-1} < \hat{m}_{000}^{(P_n^w)}$ and $m_{i-2} > \hat{m}_{000}^{(P_n^w)}$, then $k_s \leftarrow k_s/2$ and $s \leftarrow s + k_s$;
 - if $m_{i-1} > \hat{m}_{000}^{(P_n^w)}$ and $m_{i-2} > \hat{m}_{000}^{(P_n^w)}$, then $s \leftarrow s - k_s$;
 - if $m_{i-1} > \hat{m}_{000}^{(P_n^w)}$ and $m_{i-2} < \hat{m}_{000}^{(P_n^w)}$, then $k_s \leftarrow k_s/2$ and $s \leftarrow s - k_s$.
 - 5: **for** each modifiable vertex v_k in P_n^w **do**
 - 6: Derive its deformation factor s_{v_k} (Eqs. (14) and (15)) according to s and its normalized cylindrical coordinates (Eqs. (12) and (13))
 - 7: Modify its original Cartesian coordinates to obtain a candidate displaced vertex $v_{k'} = (x_{k'}, y_{k'}, z_{k'})$ by using the following rule:
 - if all the incident facets of v_k have positive moment contributions, then $(x_{k'}, y_{k'}, z_{k'}) = s_{v_k} \cdot (x_k, y_k, z_k)$;
 - if all the incident facets of v_k have negative moment contributions, then $(x_{k'}, y_{k'}, z_{k'}) = (2 - s_{v_k}) \cdot (x_k, y_k, z_k)$.
 - 8: **end for**
 - 9: set m_i as the zero-order volume moment of the obtained deformed patch
 - 10: iteration number incrementation: $i \leftarrow i + 1$
 - 11: **until** $|m_i - \hat{m}_{000}^{(P_n^w)}| < \varepsilon$ or $i = I_{max}$

Fig. 5 illustrates the distortion effects of a moderate-intensity watermark and an extremely strong-intensity watermark. There

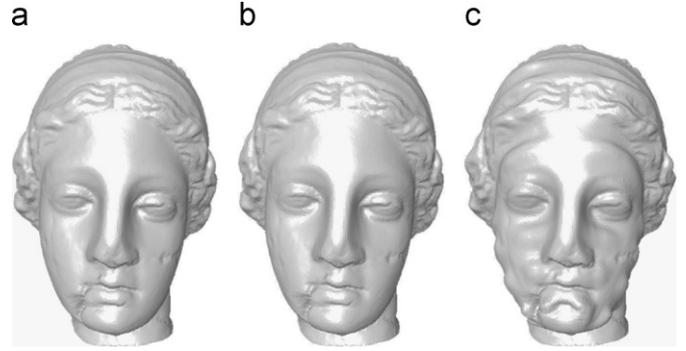


Fig. 5. This figure illustrates the visual effects of the patch deformation: (a) the original Venus; (b) a moderately watermarked Venus; and (c) a strongly watermarked Venus.

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. In practice, we never use a strong embedding strength as that illustrated in Fig. 5(c) to watermark a 3D mesh; a moderate strength as in Fig. 5(b) already leads to a satisfactory robustness.

4.7. Moment compensation

The objective of this step is to recover the original center position and principal axis orientations of the cover mesh so as to resolve the causality problem. Concretely, we need 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 induced by the cover patch deformation in the last step, so that these six moments all become zero again, or at least reasonably small. After this moment compensation, we can recover the canonical pose of the original normalized mesh at the extraction phase, and thus correctly decompose the model and extract the embedded bits.

As mentioned in Section 4.4, the 12 patches having the highest m_{000} amplitudes are chosen to accomplish the moment compensation task. In our experiments, we fix the number of compensation patches as 12 since it ensures a quite successful compensation on most meshes. In the future, we would like to find adaptive and optimum numbers of compensation patches for individual meshes with different sizes and shape complexities, if there exist. Our compensation method is based on the following property of the iterative patch deformation process described in Algorithm 1: when deforming a patch by using this algorithm, it can be proven (cf. Appendix B) and has also been experimentally validated that the moment variation ratios $\Delta m_{100}^{(P_j)} / \Delta m_{000}^{(P_j)}$, $\Delta m_{010}^{(P_j)} / \Delta m_{000}^{(P_j)}$, $\Delta m_{001}^{(P_j)} / \Delta m_{000}^{(P_j)}$, $\Delta m_{110}^{(P_j)} / \Delta m_{000}^{(P_j)}$, $\Delta m_{101}^{(P_j)} / \Delta m_{000}^{(P_j)}$ and $\Delta m_{011}^{(P_j)} / \Delta m_{000}^{(P_j)}$ are kept approximately constant under different values of the global deformation factor s . The compensation patches $P_l^c, l \in \{0, 1, \dots, 11\}$ are deformed arbitrarily by using Algorithm 1 prior to the moment compensation step so as to learn the values of these ratios (the 12 patches are then restored to their initial shape). For the sake of notation simplicity, the six learned ratios of the compensation patch P_l^c are hereafter denoted by r_1^l to 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 moments of the other orders 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 the watermark bit embedding through cover patch deformation. A 6×12 linear

least-squares system is constructed:

$$\tilde{M} = \underset{M}{\operatorname{argmin}} \|R.M - \tilde{M}\|_2^2, \quad (16)$$

where R is a 6×12 matrix with $R_{ij} = r_i^{j-1}$, M is a 12×1 matrix with $M_{i1} = \Delta m_{000}^{i-1}$, and $\tilde{M} = [\tilde{m}_{100} \ \tilde{m}_{010} \ \tilde{m}_{001} \ \tilde{m}_{110} \ \tilde{m}_{101} \ \tilde{m}_{011}]^T$. The optimization of the above system is subject to two constraints:

$$Lb \leq M \leq Ub, \quad (17)$$

$$R.M' = \tilde{M}', \quad (18)$$

where Lb and Ub represent, respectively, 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 amount of deformation. We have selected some appropriate values for the lower and upper bounds so that the deformation amplitude of the compensation patches is of 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 observation that our whole watermarking algorithm is experimentally much more sensitive to the principal axis orientation change than to the mesh center change.

We solve the least-squares system established in Eq. (16) subject to the two constraints expressed in Eqs. (17) and (18), and obtain the correct moment variations (thus the target zero-order moment values) for the 12 compensation patches. These patches are afterwards deformed by using Algorithm 1 so as to attain the wanted moment values. After this step, the six compensated first- and second-order moments of the obtained mesh are very close to zero and normally will not have negative influence on the blind watermark bit extraction. The last step of the watermark embedding procedure is the removal of the auxiliary vertices and edges that were inserted during the patch decomposition step.

5. Watermark extraction

The watermark extraction algorithm (cf. Fig. 1(b)) is blind and fast. First, the input mesh is normalized by using the technique

described in Section 4.2. Then, the vertex coordinates are converted into cylindrical system and the mesh is decomposed into patches by discretizing its h and θ domains. After using the patch classification rules presented in Section 4.4, we can pick out the candidate cover patches for the watermark bit extraction. Next, with the knowledge of the secret key K and by using Eqs. (8) to (10)), we construct a codebook $\hat{U}_{m_{000}^{(P_n^W)}, t^{(P_n^W)}}$ for each cover patch.

According to the actual moment value $m_{000}^{(P_n^W)}$ of the patch, we can find the codeword $\hat{u}_{m_{000}^{(P_n^W)}}$ that is the closest to $m_{000}^{(P_n^W)}$ in the codebook.

Finally, the n -th extracted watermark bit w_n is evaluated as the represented bit a of the retrieved codeword $\hat{u}_{m_{000}^{(P_n^W)}}$.

6. Results and comparisons

6.1. Baseline evaluation

The proposed method is implemented in C++, and has been tested on several mesh models. Fig. 6 illustrates four of them: Venus (100 759 vertices), Horse (112 642 vertices), Bunny (34 835 vertices) and Dragon (50 000 vertices). Fig. 7 illustrates the watermarked meshes (also called *stego* models in watermarking terminology). We can see that the induced distortion is quite imperceptible, 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 sensitive [7,8]. Fig. 8 illustrates the maps of the geometric objective distortions between the original and the watermarked meshes. It can be noticed that although the distortions are globally well balanced, there still exist some patches which are much more deformed than the others. More precisely, bumps and concave parts may be perceived on the mesh surface if the watermarking intensity is relatively strong, such as in Fig. 7. According to our experiments, the locations of these bumps and concave parts are rather random if we modify the secret key K . Therefore, this drawback is rather due to the moment quantization

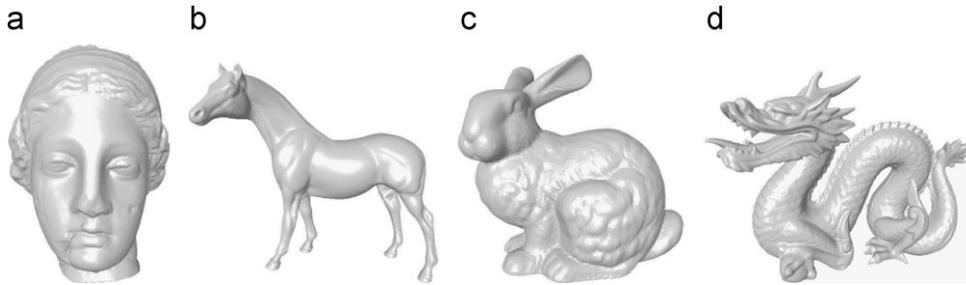


Fig. 6. The original non-watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, and (d) Dragon.

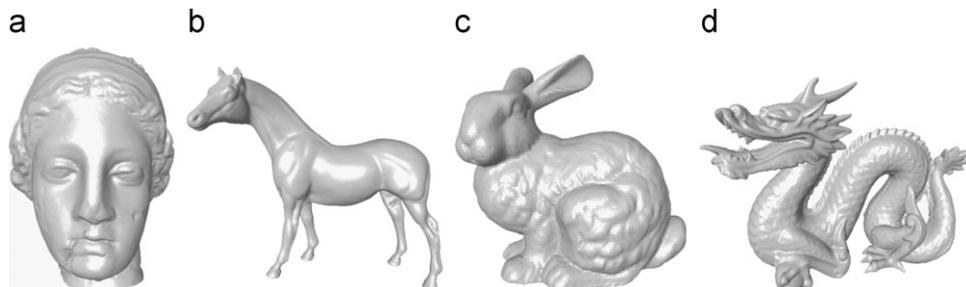


Fig. 7. The watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, and (d) Dragon.

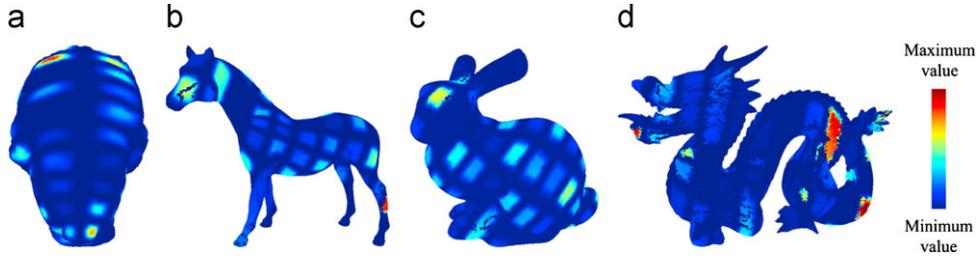


Fig. 8. The objective distortion maps of the watermarked meshes: (a) Venus, (b) Horse, (c) Bunny, and (d) Dragon.

Table 2
Baseline evaluations of the proposed watermarking method.

Mesh model \Rightarrow	Venus	Horse	Bunny	Dragon
Embedding time (s)	410.8	191.5	109.4	166.2
Extraction time (s)	3.2	2.9	1.1	1.6
WM payload (bit)	75	46	67	49
MRMS by WM (10^{-3})	2.34	1.04	1.75	1.76
MSDM by WM	0.15	0.17	0.19	0.20

scheme, than to the masked deformation algorithm or the local property of the patch. The understanding and improvement of this point constitutes one important part of our future work.

Table 2 details some statistics about the watermark embedding and extraction algorithms. All the tests were carried out on a laptop equipped with a Pentium IV 2.0GHz processor and 2GB memory. The objective distortion between the normalized cover and stego meshes are measured by Metro [34] 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 [35]: Its value tends towards 1 (theoretical limit) when the measured objects are visually very different and is equal to 0 for identical ones. One advantage of our method is that it can introduce relatively high-amplitude deformation while keeping it imperceptible. Most of the embedding time is spent on the iterative deformation step, which depends not only on the size of the mesh (i.e. its vertex number) but also on its cover patch number. The extraction time is almost completely due to the patch decomposition operation and is basically proportional to the mesh size.

6.2. Robustness evaluation

The resistance of the embedded watermark has been tested under different types of attacks. The robustness is evaluated in terms of the BER (bit error rate) of the extracted watermark bit sequence, as well as the correlation coefficient [2] between the extracted watermark bit string $\{w'_n\}$ and the originally inserted one $\{w_n\}$ as given by the following equation:

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

where \bar{w}' and \bar{w} indicate, respectively, the averages of the watermark bit strings $\{w'_n\}$ and $\{w_n\}$. This correlation value measures the similarity between two strings and varies between -1 (orthogonal strings) and $+1$ (the same strings). The distortions induced by the attacks are measured by MRMS.

6.2.1. Robustness against geometry attacks

First, our watermark is experimentally invariant to the so-called *content preserving attacks* including vertex/facet reordering in the mesh file and similarity transformation (i.e. translation,

Table 3
Robustness against random noise addition.

Model	Amplitude (%)	MRMS (10^{-3})	BER	Correlation
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 4
Robustness against Laplacian smoothing ($\lambda = 0.03$).

Model	Iteration	MRMS (10^{-3})	BER	Correlation
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

rotation, uniform scaling and their combination). As for the other geometry attacks, Tables 3, 4 and 5, respectively, present the robustness evaluation results under noise addition, smoothing and uniform coordinate quantization. Some geometrically attacked models are illustrated in Fig. 9(a)–(d). The maximum amplitude A of the random additive noise is relative to the average distance from the vertices to the mesh center. The actual noise amplitudes on the individual vertex coordinates are pseudo-random values uniformly distributed in the interval $[-A, A]$. For each amplitude level A , 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. In smoothing attacks, the mesh is processed by Laplacian smoothing [36] with different iteration numbers while fixing the deformation factor λ as 0.03. In quantization attacks, the vertex coordinates are uniformly quantized: an 8-bit quantization means that each coordinate is rounded to one of the 256 possible levels. Our algorithm demonstrates a fairly high robustness against geometry attacks, even those with strong amplitudes and those that are spatially non-uniform (anisotropic). For instance, in average, we can still successfully extract up to 84% of the watermark bits under 0.50% noise addition (the visual effect of this attack is illustrated in Fig. 9(a)). The watermarks embedded in Bunny and Dragon are

Table 5
Robustness against uniform quantization of the vertex coordinates.

Model	Intensity	MRMS (10^{-3})	BER	Correlation
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

less robust against smoothing because this attack induces an important shrinking effect on these two models.

6.2.2. Robustness against connectivity attacks

The tested connectivity attacks include surface simplification (spatially uniform and non-uniform), subdivision and remeshing. For surface simplification, we use Garland and Heckbert's quadric-error-metric-based method [37], combined with different vertex reduction ratios. The subdivision attacks include the simple midpoint scheme, the modified butterfly scheme and the Loop scheme [38]. The remeshing attack is a uniform resampling of the mesh vertices using the ReMESH software [39]; two different target vertex numbers are considered: they are, respectively, 100% and 50% of the original vertex number of the watermarked mesh.

Tables 6–8 present the corresponding robustness evaluation results. In Fig. 9(e)–(h), some attacked models are illustrated. It can be observed that our scheme has a very strong robustness against all these connectivity attacks, which are in general considered very difficult to handle for a blind mesh watermarking algorithm. As an example, for Venus and Horse, we can still retrieve 93% of the watermark bits after having removed 97.5% of the vertices in the models. The watermark embedded in Dragon is less robust against connectivity attacks since the model has a relatively low number of vertices regarding its complexity, therefore modifying its connectivity induces an important modification on the model's 3D shape.

6.2.3. 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

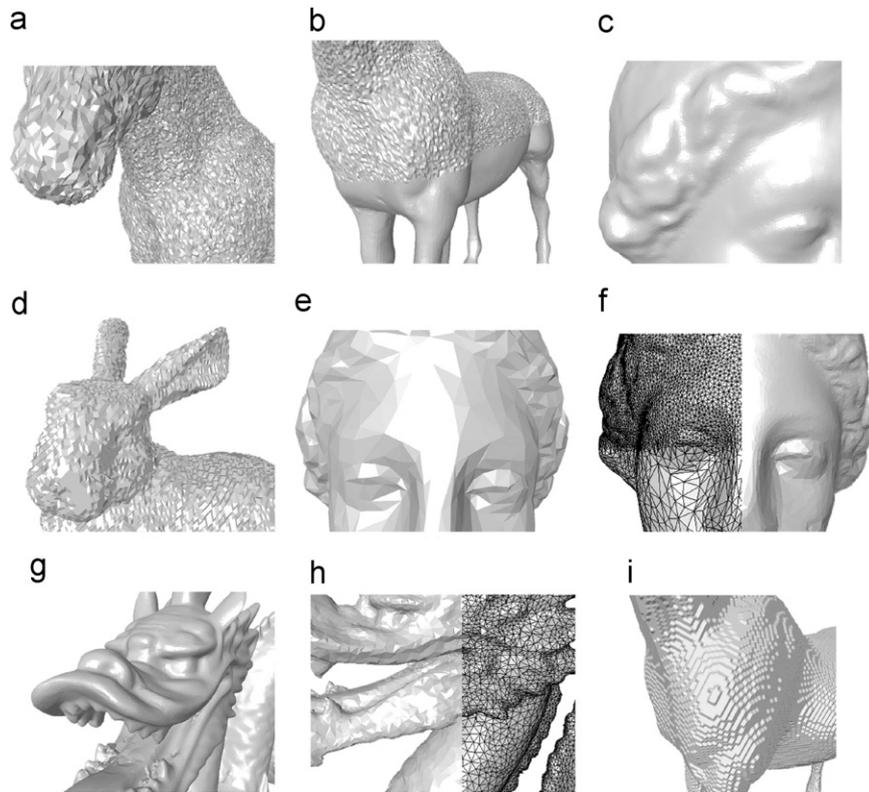


Fig. 9. Close-ups of some attacked watermarked models: (a) 0.50% random additive noise (BER=0.12); (b) 0.50% spatially non-uniform noise (BER=0.11); (c) 100-iteration Laplacian smoothing with $\lambda = 0.03$ (BER=0.08); (d) 7-bit coordinate quantization (BER=0.15); (e) spatially uniform simplification by 97.5% vertex reduction (BER=0.07); (f) spatially 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); and (i) output mesh of the marching cubes algorithm on a $350 \times 350 \times 350$ discretized (voxelized) Horse (BER=0.11).

Table 6
Robustness against surface simplification.

Model	Vertex reduction ratio (%)	MRMS (10^{-3})	BER	Correlation
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

Table 7
Robustness against one-step subdivision.

Model	Scheme	MRMS (10^{-3})	BER	Correlation
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 8
Robustness against uniform surface remeshing.

Model	Vertex number (%)	MRMS (10^{-3})	BER	Correlation
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

grid. In order to extract the watermark from this discrete volumetric representation, we transform it back to a mesh representation by using the well-known marching cubes algorithm [40]. The watermark extraction is then carried out on this reconstructed mesh. Table 9 presents the robustness results under this attack. For Venus, Horse and Bunny, the robustness is quite satisfactory (BER is around 0.12), considering the high strength of this attack (cf. Fig. 9(i)). The watermark extraction on Dragon fails because the marching cubes algorithm has created very strong

Table 9
Robustness against voxelization.

Model	Resolution	MRMS (10^{-3})	BER	Correlation
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

artefacts on its tail, which significantly change the mesh's center and principal axes.

6.3. Discussion and comparison with the methods of Cho et al.

In this subsection, we provide some discussions on the strengths and shortcomings of the proposed watermarking method. We will also compare our method with the two methods from Cho et al. [24], which are considered as the most robust blind mesh watermarking algorithms proposed so far. 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 of all, concerning the watermark imperceptibility, the induced patch deformation in our scheme is of low frequency while their methods seem to introduce relatively high-frequency distortions. Fig. 10 illustrates the Horse and Bunny models watermarked by their and our methods. The objective MRMS distances introduced by their watermark embedding (0.51×10^{-3} for Horse and 0.29×10^{-3} for Bunny) are smaller, but these small-amplitude objective distortions seem to be more perceptible (cf. Fig. 10(a) and (c)). This point is also confirmed by the MSDM perceptual distances between their watermarked and original models (0.23 for Horse and 0.32 for Bunny against, respectively, 0.17 and 0.19 for models watermarked by our method). In particular, some ring-like high-frequency artefacts may occur on the surface of their watermarked meshes, especially on smooth regions like the body of Horse.

Tables 10 and 11 present the robustness evaluation results of the watermarks embedded in the stego models of Cho et al. that are illustrated in Fig. 10(a) and (c). For comparison, the corresponding correlation values of our method are also listed in the last columns of these tables. The robustness comparison was carried out under the premise of a same watermarking payload. From these results, we can see that our stego Horse that has nearly no visual distortion is more robust, under both geometry and connectivity attacks, than their stego Horse on which there exist noticeable distortions. In particular, our algorithm is more resistant against quantization (our correlation is 1 against 0.66 for their method, under a 9-bit quantization) and simplification (1 against 0.58 for correlation values under a 90% simplification), which constitute the two most realistic attacks on watermarked meshes. Our stego Bunny has also a better imperceptibility than theirs, and is more robust against connectivity attacks (especially under surface simplification). Robustness against geometry attacks is quite similar: our algorithm is globally more robust to strong distortions while their method performs better against small-amplitude attacks. One exception is the smoothing attack which induces obvious shrinkage deformations on this relatively sparse surface and thus leads to a bad performance of our method. In general, their methods have difficulties under strong-amplitude non-uniform simplifications since the calculated mesh center can be mistakenly moved towards the mesh part where the vertex density is higher.

In all, our method is particularly suitable for the protection of dense meshes, for which the imperceptibility and the robustness

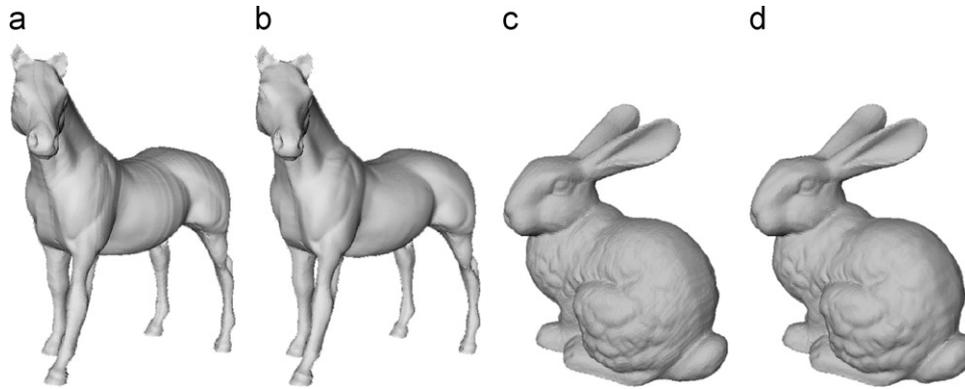


Fig. 10. Imperceptibility comparison between the algorithms of Cho et al. [24] and our method: (a) Horse watermarked by their Algorithm I (strength parameter $\alpha = 0.03$); (b) Horse watermarked by our method; (c) Bunny watermarked by their Algorithm II (strength parameter $\alpha = 0.07$); and (d) Bunny watermarked by our method.

Table 10

Robustness evaluation results for the Horse model watermarked by Algorithm I of Cho et al. [24] ($\alpha = 0.03$, 46 bits are embedded), in comparison with our method.

Attack	Cho's BER	Cho's 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 11

Robustness evaluation results for the Bunny model watermarked by Algorithm II of Cho et al. [24] ($\alpha = 0.07$, 67 bits are embedded), in comparison with our method.

Attack	Cho's BER	Cho's 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
100% uniform remeshing	0.02	0.97	0.94
50% uniform remeshing	0.22	0.57	0.74

against simplification are the main concerns. The advantage of the algorithms of Cho et al. is that the watermark can resist attacks that introduce much higher objective distortions than its embedding. Neither the methods of Cho et al. nor our method achieves

the robustness against strong local deformation and cropping due to the de-synchronization problem.

One drawback of our watermarking scheme is that its payload depends on the mesh shape and normally varies from 45 to 75 bits. The payload actually depends on the number of cover patches in the mesh, and thus varies from model to model. However, it is not necessary to transmit the number of embedded bits to the extraction side, because the extractor can automatically determine this number as $(N - 1)$, where N is the number of cover patches in the input model. On the contrary, the payload of the methods of Cho et al. is independent from the specific shape of the cover mesh and thus can ensure a constant value, say 64 bits, for all the 3D models.

Another observation is that our method works better on meshes with shapes somehow similar to a cylinder such as Venus and Horse, than on those with complex shapes such as Dragon, in terms of both payload and robustness. The reason is that the proposed cylindrical patch decomposition may produce degenerate patches on those complex models. As for the payload, some small-sized patches (i.e. discarded patches) will be generated in which we cannot embed watermark bits (nonetheless, at present we can normally ensure a payload of at least 40 bits, even on very complex models). As for the robustness, some of the obtained cover patches are ill-shaped, and experimentally their moment values are not as robust as the "regular" patches with roughly a curved-square shape. Ideally, this problem would be resolved if we could devise an adaptive and robust mesh decomposition algorithm which produces a fixed number of well-shaped patches on arbitrary models. But unfortunately, the development of such a method seems an open problem.

Finally, although our main objective is to achieve a strong robustness against connectivity attacks, it seems that the proposed method ensures a minimum level of watermarking security, which is a rather high-level requirement [41,42] and is often omitted in mesh watermarking research whose main concern is still the robustness. It seems difficult for a pirate to carry out unauthorized watermark detection or optimal watermark removal if he does not have the secret key K , even if he knows the details of the watermarking method. The reason is that the pirate will have difficulties in correctly constructing the pseudo-random codebooks for the cover patch volume moments.

6.4. Evaluation and comparison within a mesh watermarking benchmark

In this subsection, we evaluate the proposed method and also compare it with more other methods within the framework of a 3D mesh watermarking benchmark recently proposed by Wang

Table 12

Baseline evaluation results of the first group of test (on Venus model, with a payload of 64 bits).

Methodology	Perceptual-quality-oriented protocol			Geometric-quality-oriented protocol		
	Wavelet [20]	Histogram [24]	Moment (this paper)	Wavelet [20]	Histogram [24]	Moment (this paper)
Watermarking method						
WM payload (bits)	64	64	64	64	64	64
Embedding time (s)	12.8	7.6	439.9	12.6	11.6	377.6
Extraction time (s)	4.9	< 1.0	3.3	4.7	< 1.0	3.5
d_{MRMS} (w.r.t. l_{bbd}) (%)	0.078	0.0080	0.069	0.019	0.012	0.018
d_{MSDM}	0.10	0.19	0.14	0.05	0.29	0.09

Table 13

Global robustness comparison of the first group of test (on Venus model, with a payload of 64 bits), in terms of BER.

Methodology	Perceptual-quality-oriented protocol			Geometric-quality-oriented protocol		
	Wavelet [20]	Histogram [24]	Moment (this paper)	Wavelet [20]	Histogram [24]	Moment (this paper)
Watermarking method						
Average BER under geometry attacks	0.14	0.18	0.05	0.21	0.14	0.24
Average BER under connectivity attacks	N.A.	0.27	0.12	N.A.	0.21	0.16
Average BER under all attacks	0.14	0.22	0.08	0.21	0.17	0.20

The lower the BER is, the more robust the method is.

et al. [43].¹ The objective of this benchmark is to facilitate the experimental comparison between different methods by normalizing the test models, the evaluation metrics, the conducted attacks and the assessment methodology. First, the benchmark contains a “standard” collection of mesh models on which we should test the watermarking methods. It also proposes to use, respectively, MRMS [34] and MSDM [35] to measure the geometric and the perceptual distortions induced by the watermark embedding. The benchmark also suggests a group of attacks against which we should test the algorithms’ robustness. Finally, two applications-oriented evaluation protocols have been established, which indicate the main steps to follow when conducting the evaluation experiments.

Similar to the Stirmark protocol [44] designed for image watermarks, the basic idea of the two mesh watermarking evaluation protocols is to first prescribe the watermark payload and the watermark induced distortions (both geometric and perceptual), and then to evaluate the robustness of the method (e.g. in terms of BER) under a series of attacks. The main difference between the two protocols is that they have very different thresholds on the induced geometric and perceptual distortions for the methods under evaluation. More precisely, the *perceptual-quality-oriented* protocol requires that the MRMS distortion d_{MRMS} should be less than 0.08%. l_{bbd} (l_{bbd} being the diagonal length of the mesh’s bounding box), and that the MSDM distortion d_{MSDM} should be no more than 0.20. In the *geometric-quality-oriented* protocol, these two thresholds are, respectively, equal to 0.02%. l_{bbd} and 0.30.

Besides our moment-based method and the histogram-based method of Cho et al. (Algorithm I) [24], the methods under comparison also include the wavelet-based method of Wang et al. [20] and the spectral-domain-based method of Wang et al. [45]. In the wavelet-based method, watermark bits are embedded through scalar quantization of the norms of the wavelet coefficient vectors associated to the coarsest resolution of the cover semi-regular mesh. In the spectral-domain-based method, the manifold harmonics spectral amplitudes [46] of the cover mesh are iteratively quantized to embed the watermark. In the following, we will briefly present three groups of testing results of these four methods obtained by using the benchmark. The tests

were carried out on different meshes, under different protocols and with different watermarking payloads.

6.4.1. Test of 64-bit schemes on Venus model

In the first group of test, we compare the wavelet-based method [20], the histogram-based method [24], and the proposed moment-based method, under both evaluation protocols. The tests were performed on the Venus model (100 759 vertices), and the methods’ payloads were all fixed as 64 bits. Table 12 presents the baseline evaluation results of the tested watermarking schemes. Like in Section 6.1, the algorithm execution times are those obtained on a laptop equipped with a 2.0 GHz processor and 2 GB memory (the same for the subsequent tests). In Table 13, we provide the average BER values of the tested methods under the geometry attacks, the connectivity attacks and all the attacks suggested by the benchmark. All the results are the averages of 5 trials with randomly selected watermark sequences and keys. In order to apply the wavelet-based watermarking method, the original irregular Venus model is remeshed prior to the watermark embedding. The average BER values in Table 13 can be considered as the benchmarking scores of the tested methods. The lower these scores are, the more robust the watermarking method is. Therefore, with these average BER values we can provide a quick and global comparison of the methods’ robustness against different kinds of attacks.

According to the execution times presented in Table 12, the histogram-based method and the wavelet-based method are much faster than the moment-based method. However, the processing time of the last method is already considered as acceptable in most of the watermarking-based applications, except for those requiring real-time watermark embedding and extraction. From the robustness evaluation results, we can conclude that, for the Venus model, the moment-based method is the most suitable method to be used in applications that require a high visual quality of the watermarked object (i.e. under the perceptual-quality-oriented protocol) since it has the lowest average BER score 0.08, while the histogram-based method is the most appropriate scheme for the applications which can only tolerate a very small amount of induced geometric distortion (i.e. under the geometric-quality-oriented protocol) since it has the lowest average BER score 0.17. However, in both kinds of applications, if a strong robustness against connectivity attacks

¹ The benchmark is freely available at <http://liris.cnrs.fr/meshbenchmark/>.

is required, then the moment-based method seems the best choice, since it has the lowest BER scores 0.12 and 0.16 under connectivity attacks, respectively, for the two protocols. The wavelet-based method shows satisfactory and roughly comparable robustness performances (against geometry attacks) under both evaluation protocols. This is somewhat surprising considering that the watermark induced distortion under the geometric-quality-oriented protocol has been significantly decreased compared to that under the perceptual-quality-oriented protocol. For the wavelet-based method, we do not provide its robustness results against connectivity attacks since these attacks in general destroy the semi-regular connectivity (and also the intrinsic attractiveness) of the stego mesh and thus makes it impossible to perform wavelet decomposition on the attacked models.

Fig. 11 illustrates the stego models obtained by the three methods. It provides a direct impression of the deformation patterns induced by the different watermark embeddings. This figure further confirms the good imperceptibility of the proposed moment-based method. In addition, we can see that the bumps and concave parts, which may be visible under a relatively strong embedding (cf. Fig. 7), can in general be avoided if we reduce the strength (cf. Fig. 11(g)), but at the expense of a diminution of robustness.

In all, the advantage of the histogram-based method is that with a very low geometric distortion induced by the watermark embedding, it can, however, resist very strong-amplitude attacks, and the main strengths of the moment-based method are its strong robustness against connectivity attacks and its high watermark imperceptibility. The wavelet-based method also has the advantage of being highly imperceptible. In addition, it seems that the robustness performance of this method is not quite dependent on the embedding strength (especially under small- and moderate-amplitude attacks; however, we have not provided these detailed results here). This implies that the selected watermarking primitive, which can be equivalently considered as the ratio between the norm

of a wavelet coefficient vector and the average length of the edges in the coarsest-level resolution (cf. [20] for more details), may be a robust geometric feature of the cover mesh.

6.4.2. Test of 16-bit schemes on Rabbit model

The second group of test was carried out on the Rabbit model (70 658 vertices). The tested methods include the spectral-domain-based method [45], the histogram-based method [24], and the proposed moment-based method, all with a payload of 16 bits. Table 14 presents the baseline evaluation results, and Table 15 presents the robustness evaluation results.

The moment-based method seems to have the best robustness, since it has the lowest overall average BER values under both protocols (respectively, equal to 0.06 and 0.12). This method is particularly robust against simplification and subdivision: actually we can always correctly extract the embedded watermark under the simplification and subdivision attacks suggested by the benchmark, without any bit error. The histogram-based method has a very good performance under geometry attacks. However, compared to the moment-based method, their method is less robust against connectivity attacks. The spectral-domain-based method has a relatively satisfactory robustness under the perceptual-quality-oriented protocol. Under the geometric-quality-oriented protocol, it has a very poor performance. A stronger robustness can be attained if we increase the watermark embedding strength for the spectral-domain-based method. As shown in [45, Tables 1 and 2], a much better robustness is achieved if the watermark induced MRMS distortion attains 2.37×10^{-3} , i.e. $0.13\% I_{bbd}$. Indeed, this is a common problem for the mesh watermarking schemes that are based on the manifold harmonics spectral transform [46] because it is difficult to precisely control the amount of induced distortion mainly due to the causality problem encountered by these schemes (cf. Ref. [45] for more details).

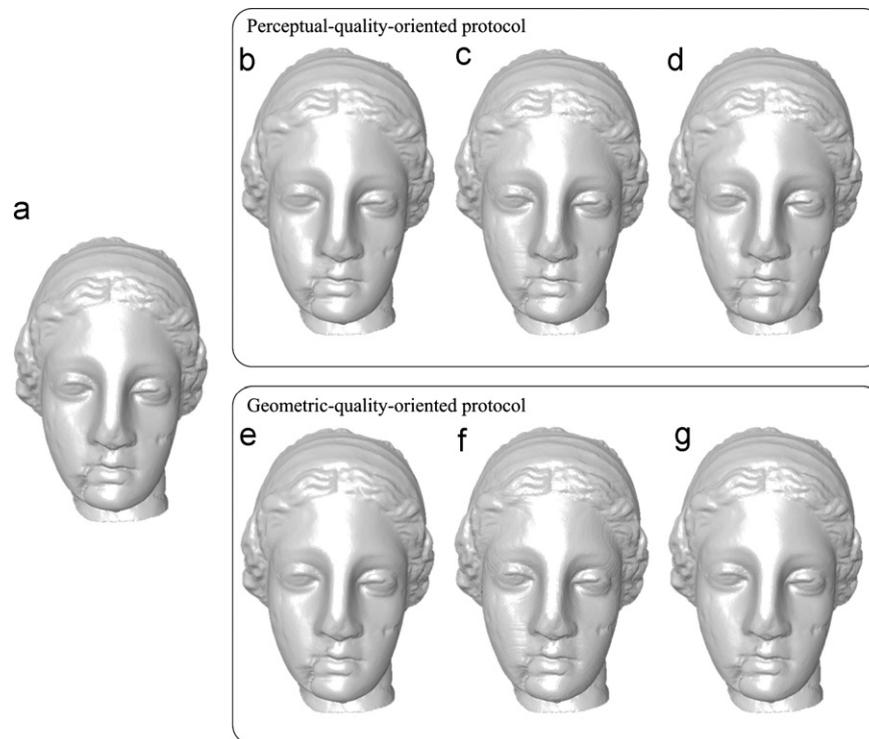


Fig. 11. Illustration of the original and watermarked meshes in the first group of test: (a) the original Venus; (b), (c) and (d) are, respectively, the watermarked meshes by using the wavelet-based method [20], the histogram-based method [24], and the moment-based method, for the perceptual-quality-oriented protocol; (e), (f) and (g) are, respectively, the watermarked meshes by using the wavelet-based method [20], the histogram-based method [24], and the moment-based method, for the geometric-quality-oriented protocol.

Table 14

Baseline evaluation results of the second group of test (on Rabbit model, with a payload of 16 bits).

Methodology	Perceptual-quality-oriented protocol			Geometric-quality-oriented protocol		
	Spectral [45]	Histogram [24]	Moment (this paper)	Spectral [45]	Histogram [24]	Moment (this paper)
Watermarking method						
WM payload (bits)	16	16	16	16	16	16
Embedding time (s)	311.5	2.7	147.2	459.4	1.5	132.0
Extraction time (s)	47.6	< 10	3.0	49.0	< 10	3.0
d_{MRMS} (w.r.t. I_{bbd}) (%)	0.064	0.058	0.067	0.017	0.020	0.018
d_{MSDM}	0.09	0.19	0.12	0.06	0.08	0.09

Table 15

Global robustness comparison of the second group of test (on Rabbit model, with a payload of 16 bits), in terms of BER.

Methodology	Perceptual-quality-oriented protocol			Geometric-quality-oriented protocol		
	Spectral [45]	Histogram [24]	Moment (this paper)	Spectral [45]	Histogram [24]	Moment (this paper)
Watermarking method						
Average BER under geometry attacks	0.22	0.07	0.02	0.37	0.06	0.13
Average BER under connectivity attacks	0.24	0.20	0.11	0.32	0.23	0.11
Average BER under all attacks	0.23	0.13	0.06	0.35	0.14	0.12

The lower the BER is, the more robust the method is.

6.4.3. ROC performance of the proposed method

In this subsection, we evaluate the proposed moment-based method by considering it as a *detectable* watermarking scheme (also called 1-bit scheme), still by using the mesh watermarking benchmark proposed in [43]. In the modified detectable scheme, after the bit extraction stage, we first calculate the correlation coefficient by using Eq. (19). Then, the obtained correlation coefficient value $Corr$ is compared with a threshold Thr in order to make the decision on the presence of the tested watermark (w_n) in the input mesh: if $Corr \geq Thr$, then the watermark is detected; otherwise the watermark is not detected.

The receiver operating characteristics (ROC) curve is a standard metric to evaluate the performance of detectable watermarking schemes, and it is also the evaluation metric suggested by the mesh watermarking benchmark. Basically, a ROC curve describes the relationship between the *false positive rate* against the *false negative rate* of the watermark detection algorithm under different values of the threshold Thr that is used for deciding the watermark presence. The false positive rate represents the probability of the bad decision in which a watermark is detected whereas it does not exist in the content. On the contrary, the false negative rate indicates the probability of the bad decision in which a watermark is not detected whereas it does exist in the content.

In general, in order to plot the ROC curves, we have to first prepare at least 50 stego models of the same object for the watermarking method under evaluation, using different random watermarks and random secret keys. We then conduct different attacks on these stego models. For each attacked model, two detections are performed: one with the right watermark and the right key, and the other with a wrong watermark and a wrong key. Afterwards, for each kind of attack of a certain strength, the false positive and false negative curves are drawn by varying the correlation threshold value Thr . These curves are then approximated by using Gaussian models and the ROC curves that represent the relationship between the false negative rate P_{fn} and the false positive rate P_{fp} are therefore obtained. The ROC curve point on which $P_{fn} = P_{fp}$ is called the equal error rate (EER) point. The P_{fn} (or the P_{fp}) value on this point, i.e. the EER, is commonly used as a brief quantitative evaluation metric of the method's ROC performance under a certain attack. The lower the EER is, the better the algorithm's performance is.

Table 16

Equal error rate (EER) values of the proposed moment-based method under two typical attacks: random noise addition and simplification (on Venus model, under the perceptual-quality-oriented protocol).

Attack	Equal error rate (EER)
0.05% noise addition	2.89×10^{-20}
0.10% noise addition	3.11×10^{-16}
0.30% noise addition	4.84×10^{-9}
0.50% noise addition	1.08×10^{-6}
30% simplification	1.01×10^{-35}
50% simplification	5.17×10^{-30}
70% simplification	1.15×10^{-25}
90% simplification	2.74×10^{-16}
95% simplification	1.00×10^{-11}
97.5% simplification	6.61×10^{-8}

The proposed method has been modified to become a detectable watermarking scheme.

Table 16 presents the EER values of the proposed moment-based method under two typical attacks: random noise addition and simplification (on Venus model, under the perceptual-quality-oriented protocol). Fig. 12 illustrates the corresponding ROC curves. From these experimental results, we can conclude that in general, the proposed method has a very satisfying ROC performance under these two typical attacks. For instance, under a 95% simplification (a very strong-amplitude attack), an appropriate threshold value can be found so that the false positive and the false negative probabilities are both equal to 1.00×10^{-11} , a very low value.

7. Conclusion and future work

In this paper, a new robust and blind 3D mesh watermarking algorithm is proposed. The watermark bits are embedded by slightly deforming some selected cover patches obtained after a simple mesh decomposition in the cylindrical coordinate system. Watermark imperceptibility is ensured by using a smooth low-frequency mask to modulate the patch deformation; besides, the causality problem is clearly resolved by introducing a geometric compensation post-processing after the watermark bit embedding. The robustness of this approach is due to the high stability of the

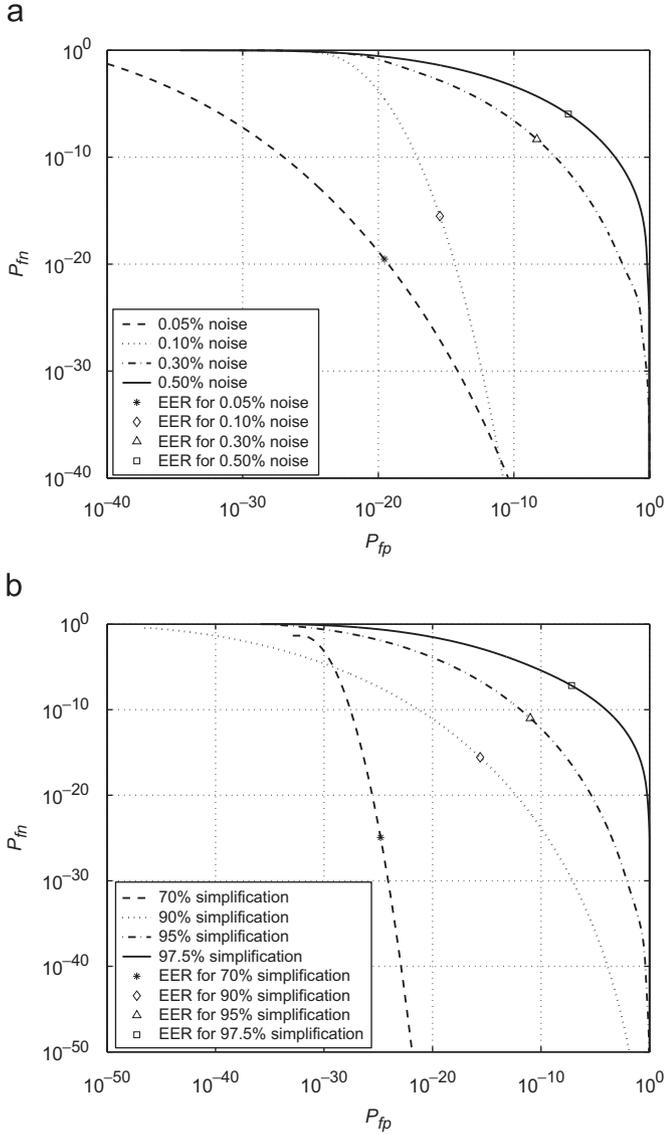


Fig. 12. Receiver operating characteristics (ROC) curves of the proposed moment-based method under (a) random noise addition attack and (b) simplification attack (on Venus model, under the perceptual-quality-oriented protocol). The proposed method has been modified to become a detectable watermarking scheme.

global and local volume moment values under geometry, connectivity and even representation conversion attacks as long as they do not seriously modify the intrinsic shape of the model. As far as we know, our method is the first in the literature that uses a continuous shape descriptor as the mesh watermarking primitive.

The proposed method can be improved in several aspects. First, it would 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 of our particular interest since it may be helpful to further balance the watermarking performance in different patches and to resolve the de-synchronization problem caused by the patch classification. We are also interested in embedding watermark in volume moments by using other data hiding techniques. Actually, the parameter setting of the quantization-based technique is complicated and it is preferred to find a simple yet better alternative. In long terms, we plan to investigate the solutions to achieving robustness against cropping (or strong local deformation) combined with connectivity attacks.

Acknowledgements

We would like to thank Dr. D. Coeurjolly for his kind help on voxelization and marching cubes algorithms. This work is in part supported by China Scholarship Council of the Chinese government and French National Research Agency (ANR) through COSINUS Program (project COLLAIVIZ no. ANR-08-COSI-003).

Appendix A. Proof 1

In this proof, we will demonstrate that it is theoretically reasonable for patches with high m_{000} moment amplitudes to receive a small Δ_{pre} value.

Recall that Δ_{pre} is involved in the determination of the component-wise quantization step $\Delta^{(P_n^w)}$ in Eqs. (9) and (10) in Section 4.5.

For the sake of simplicity, we consider a simple patch composed of only one triangle facet $f = \{v_1, v_2, v_3\} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)\}$ and its uniformly scaled version $f' = \{v_1', v_2', v_3'\} = \{(k.x_1, k.y_1, k.z_1), (k.x_2, k.y_2, k.z_2), (k.x_3, k.y_3, k.z_3)\}$ ($k > 1$). We then make the assumption that the moment quantization on these two patches should introduce comparable variations on the coordinates of their comprised vertices, so as to ensure a uniform deformation. Note that here we take into account only the objective distance metric without any perceptual considerations, still for the sake of simplicity.

Now, assume that f and f' have positive zero-order moments and are subject to a same facet vertex coordinate variation $(\Delta x_1, \Delta y_1, \Delta z_1), (\Delta x_2, \Delta y_2, \Delta z_2), (\Delta x_3, \Delta y_3, \Delta z_3)$, which simulates the consequence of the watermark embedding. After neglecting the second- and higher-order terms (e.g. $\Delta x_1 \cdot \Delta y_2 \cdot \Delta z_3$ and $\Delta x_1 \cdot \Delta y_2 \cdot \Delta z_3$) in the moment calculation formulae, we can easily find out that the following relationship approximately holds:

$$\Delta m_{000}^{(f')} = k^2 \Delta m_{000}^{(f)}, \quad (20)$$

where $\Delta m_{000}^{(f')}$ and $\Delta m_{000}^{(f)}$ are, respectively, the moment variations of f' and f . Considering that $m_{000}^{(f')} = k^3 m_{000}^{(f)}$, we then obtain

$$\frac{\Delta m_{000}^{(f')}}{m_{000}^{(f')}} = \frac{1}{k} \cdot \frac{\Delta m_{000}^{(f)}}{m_{000}^{(f)}}. \quad (21)$$

The reason for neglecting the second- and higher-order terms is explained as follows. Indeed, the vertex coordinate alteration during the watermark embedding is quite small (in the order of 0.10%). Therefore, the second-order terms are much smaller than the first-order terms (also of about 0.10%). Hence, these small-value terms can be neglected in the above analysis without introducing significant errors.

From Eq. (9) which presents the calculation of the component-wise quantization step $\Delta^{(P_n^w)}$, we can see that the terms

$$\left| m_{000}^{(\hat{P}_{n-1}^w)} / \left[\frac{m_{000}^{(\hat{P}_{n-1}^w)}}{m_{000}^{(P_n^w)}} \right] \right| \quad \text{and} \quad \left| m_{000}^{(\hat{P}_{n-1}^w)} \cdot \left[\frac{m_{000}^{(P_n^w)}}{m_{000}^{(\hat{P}_{n-1}^w)}} \right] \right|$$

are approximately equal to the moment amplitude of the current patch (i.e. $m_{000}^{(f)}$ or $m_{000}^{(f')}$ in Eq. (21)). Meanwhile, the final moment variation (i.e. $\Delta m_{000}^{(f)}$ or $\Delta m_{000}^{(f')}$ in Eq. (21)) is somewhat proportional to the quantization step $\Delta^{(P_n^w)}$. Therefore (cf. Eq. (9)), the term Δ_{pre} approximately represents the ratio between the moment variation and the original moment value of the current patch. From Eq. (21), we can deduce that, in order to have comparable vertex variations for the two patches under consideration, the above mentioned ratio should be smaller for the

patch having a higher m_{000} value due to the existence of the term $1/k$ ($k > 1$) on the right side of the formula.

Although the above deduction makes many assumptions and considers only a very simple case, it potentially constitutes a reasonable proof for setting a small Δ_{pre} value for patches with high m_{000} moment amplitudes. This measure has also been demonstrated effective in practice since it can, to some extent, balance the watermark induced distortions in patches of different sizes.

Appendix B. Proof 2

In this proof, we will demonstrate the following property: when deforming a patch \mathcal{P}_j by using Algorithm 1, the moment variation ratios

$$\frac{\Delta m_{100}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}, \frac{\Delta m_{010}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}, \frac{\Delta m_{001}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}, \frac{\Delta m_{110}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}, \frac{\Delta m_{101}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}} \quad \text{and} \quad \frac{\Delta m_{011}^{(\mathcal{P}_j)}}{\Delta m_{000}^{(\mathcal{P}_j)}}$$

are kept approximately constant under different values of the global deformation factor s .

We will take $\Delta m_{100}^{(\mathcal{P}_j)}/\Delta m_{000}^{(\mathcal{P}_j)}$ as an example to carry out the demonstration and hereafter neglect the patch designation superscripts in the notations of the volume moments and their variations.

First, we can rewrite the m_{000} calculation formula for a patch composed of vertices v_1, v_2, \dots, v_N as the following sum of several multiplication terms of three vertex coordinates:

$$m_{000} = \frac{1}{6} \sum_{i,j,k} \text{sign}_{ijk} x_i y_j z_k, \quad (22)$$

where $\text{sign}_{ijk} \in \{-1, 1\}$ and the triplet $x_i y_j z_k$ occurs only if v_i, v_j and v_k are within a same facet.

For the sake of simplicity, we suppose hereafter that all the facets in the patch have positive moment contributions and $s > 1$. Under the proposed modulated patch deformation, x_i becomes $s_{v_i} x_i$ after a displacement, where $s_{v_i} = 1 + wt_{hi} \cdot wt_{\theta_i}$. It is easy to deduce (cf. Eq. (14)) that the above two weights can be rewritten as

$$wt_{hi} = a_1 \cdot \sqrt{s-1}, \quad (23)$$

$$wt_{\theta_i} = a_2 \cdot \sqrt{s-1}. \quad (24)$$

Note that a_1 and a_2 only depend on the normalized coordinates of the vertex v_i and the shape of the modulation function, which are invariant under different s values. Thus, we obtain $s_{v_i} = 1 + a_1 a_2 (s-1)$ and $x_i' = x_i + a_1 a_2 (s-1) x_i = x_i + c_i (s-1) x_i$, with $c_i = a_1 a_2$. After neglecting the second- and higher-order terms in the moment calculation formulae, we can deduce the m_{000} moment variation as follows:

$$\begin{aligned} \Delta m_{000} &= \frac{1}{6} \sum_{i,j,k} (c_i (s-1) x_i y_j z_k + c_j (s-1) x_i y_j z_k + c_k (s-1) x_i y_j z_k) \\ &= \frac{1}{6} (s-1) \sum_{i,j,k} (c_i x_i y_j z_k + c_j x_i y_j z_k + c_k x_i y_j z_k). \end{aligned} \quad (25)$$

Similarly, we can deduce the approximative variation of the moment m_{100} as:

$$\begin{aligned} \Delta m_{100} &= \frac{1}{24} \sum_{i,j,k,l} (c_i (s-1) x_i x_j y_k z_l + c_j (s-1) x_i x_j y_k z_l + c_k (s-1) x_i x_j y_k z_l + c_l (s-1) x_i x_j y_k z_l) \\ &= \frac{1}{24} (s-1) \sum_{i,j,k,l} (c_i x_i x_j y_k z_l + c_j x_i x_j y_k z_l + c_k x_i x_j y_k z_l + c_l x_i x_j y_k z_l). \end{aligned} \quad (26)$$

Hence, the ratio between Δm_{100} and Δm_{000} is equal to

$$\begin{aligned} \frac{\Delta m_{100}}{\Delta m_{000}} &= \frac{\frac{1}{24} (s-1) \sum_{i,j,k,l} (c_i x_i x_j y_k z_l + c_j x_i x_j y_k z_l + c_k x_i x_j y_k z_l + c_l x_i x_j y_k z_l)}{\frac{1}{6} (s-1) \sum_{i,j,k} (c_i x_i y_j z_k + c_j x_i y_j z_k + c_k x_i y_j z_k)} \\ &= \frac{1}{4} \cdot \frac{\sum_{i,j,k,l} (c_i x_i x_j y_k z_l + c_j x_i x_j y_k z_l + c_k x_i x_j y_k z_l + c_l x_i x_j y_k z_l)}{\sum_{i,j,k} (c_i x_i y_j z_k + c_j x_i y_j z_k + c_k x_i y_j z_k)}. \end{aligned} \quad (27)$$

Finally, we can conclude that the above moment variation ratio is completely determined by the original coordinates of the patch vertices (under a fixed modulation function shape) and thus is independent of the value of the global deformation factor s . \square

References

- [1] Botsch M, Pauly M, Kobbelt L, Alliez P, Lévy B, Bischoff S, et al. Geometric modeling based on polygonal meshes. In: Proceedings of the ACM Siggraph course notes; 2007. p. 1–181.
- [2] Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker T. Digital watermarking and steganography. 2nd ed. Morgan Kaufmann Publishers Inc.; 2007.
- [3] Barni M, Bartolini F. Watermarking systems engineering: enabling digital assets security and other applications. Marcel Dekker Inc.; 2004.
- [4] Rondao-Alfaca P, Macq B. From 3D mesh data hiding to 3D shape blind and robust watermarking: a survey. LNCS Transactions on Data Hiding and Multimedia Security 2007;2:99–115.
- [5] Wang K, Lavoué G, Denis F, Baskurt A. A comprehensive survey on three-dimensional mesh watermarking. IEEE Transactions on Multimedia 2008;10(8): 1513–27.
- [6] Bors AG. Watermarking mesh-based representations of 3-D objects using local moments. IEEE Transactions on Image Processing 2006;15(3):687–701.
- [7] Sorkine O, Cohen-Or D, Toledo S. High-pass quantization for mesh encoding. In: Proceedings of the symposium on geometry processing; 2003. p. 42–51.
- [8] Zhang H, van Kaick O, Dyer R. Spectral methods for mesh processing and analysis. In: Proceedings of the Eurographics state-of-the-art reports; 2007. p. 1–22.
- [9] Ohbuchi R, Masuda H, Aono M. Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygonal models. Computer Communications 1998;21(15):1344–54.
- [10] Cayre F, Macq B. Data hiding on 3-D triangle meshes. IEEE Transactions on Signal Processing 2003;51(4):939–49.
- [11] Agarwal P, Prabhakaran B. Robust blind watermarking mechanism for point sampled geometry. In: Proceedings of the ACM multimedia and security workshop; 2007. p. 175–86.
- [12] Wang YP, Hu SM. A new watermarking method for 3D models based on integral invariants. IEEE Transactions on Visualization and Computer Graphics 2009;15(2):285–94.
- [13] Kanai S, Date H, Kishinami T. Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In: Proceedings of the international workshop on geometric modeling; 1998. p. 296–307.
- [14] Praun E, Hoppe H, Finkelstein A. Robust mesh watermarking. In: Proceedings of the ACM Siggraph; 1999. p. 49–56.
- [15] Ohbuchi R, Mukaiyama A, Takahashi S. A frequency-domain approach to watermarking 3D shapes. Computer Graphics Forum 2002;21(3):373–82.
- [16] Wu J, Kobbelt L. Efficient spectral watermarking of large meshes with orthogonal basis functions. The Visual Computer 2005;21(8–10):848–57.
- [17] Cayre F, Rondao-Alfaca P, Schmitt F, Macq B, Maître H. Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry. Signal Processing: Image Communications 2003;18(4):309–19.
- [18] Uccheddu F, Corsini M, Barni M. Wavelet-based blind watermarking of 3D models. In: Proceedings of the ACM multimedia and security workshop; 2004. p. 143–54.
- [19] Rondao-Alfaca P, Macq B. Blind watermarking of 3D meshes using robust feature points detection. In: Proceedings of the IEEE international conference on image processing; 2005. p. 693–6.
- [20] Wang K, Lavoué G, Denis F, Baskurt A. Hierarchical watermarking of semi-regular meshes based on wavelet transform. IEEE Transactions on Information Forensics and Security 2008;3(4):620–34.
- [21] Luo M, Bors AG. Principal component analysis of spectral coefficients for mesh watermarking. In: Proceedings of the IEEE international conference on image processing; 2008. p. 441–4.
- [22] Benedens O. Geometry-based watermarking of 3D models. IEEE Computer Graphics and Applications 1999;19(1):46–55.
- [23] Zafeiriou S, Tefas A, Pitas I. Blind robust watermarking schemes for copyright protection of 3D mesh objects. IEEE Transactions on Visualization and Computer Graphics 2005;11(5):596–607.
- [24] Cho J-W, Prost R, Jung H-Y. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. IEEE Transactions on Signal Processing 2007;55(1):142–55.

- [25] Zhang C, Chen T. Efficient feature extraction for 2D/3D objects in mesh representation. In: Proceedings of the IEEE international conference on image processing; 2001. p. 935–8.
- [26] Sheynin SA, Tuzikov AV. Explicit formulae for polyhedra moments. *Pattern Recognition Letters* 2001;22(10):1103–9.
- [27] Tuzikov A, Sheynin S, Vasiliev PV. Computation of volume and surface body moments. *Pattern Recognition* 2003;36(11):2521–9.
- [28] Eggers JJ, Bauml R, Tzschoppe R, Girod B. Scalar costa scheme for information embedding. *IEEE Transactions on Signal Processing* 2003;51(4):1003–19.
- [29] Kalivas A, Tefas A, Pitas I. Watermarking of 3D models using principal component analysis. In: Proceedings of the IEEE international conference on multimedia & expo; 2003. p. 637–40.
- [30] Rondao-Alface P, Macq B, Cayre F. Blind and robust watermarking of 3D models: how to withstand the cropping attack? In: Proceedings of the IEEE international conference on image processing; 2007. p. V465–8.
- [31] Costa M. Writing on dirty paper. *IEEE Transactions on Information Theory* 1983;29(3):439–41.
- [32] Pérez-González F, Mosquera C, Barni M, Abrardo A. Rational dither modulation: a high-rate data-hiding method invariant to gain attacks. *IEEE Transactions on Signal Processing* 2005;53(10):3960–75.
- [33] Chen B, Wornell GW. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory* 2001;47(4):1423–43.
- [34] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. *Computer Graphics Forum* 1998;17(2):167–74.
- [35] Lavoué G, Gelasca ED, Dupont F, Baskurt A, Ebrahimi T. Perceptually driven 3D distance metrics with application to watermarking. In: Proceedings of the SPIE electronic imaging, vol. 6312; 2006. 63120L.1–12.
- [36] Taubin G. Geometric signal processing on polygonal meshes. In: Proceedings of the Eurographics state-of-the-art reports; 2000. p. 81–96.
- [37] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of the ACM Siggraph; 1997. p. 209–16.
- [38] Zorin D, Schröder P. Subdivision for modeling and animation. In: Proceedings of the ACM Siggraph course notes; 2000. p. 1–116.
- [39] Attene M, Falcidieno B. ReMESH: an interactive environment to edit and repair triangle meshes. In: Proceedings of the IEEE International conference on shape modeling and applications; 2006. p. 271–6.
- [40] Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. In: Proceedings of the ACM Siggraph; 1987. p. 163–70.
- [41] Comesaña P, Pérez-Freire L, Pérez-González F. Fundamentals of data hiding security and their application to spread-spectrum analysis. In: Proceedings of the international workshop on information hiding; 2005. p. 146–60.
- [42] Cayre F, Fontaine C, Furon T. Watermarking security: theory and practice. *IEEE Transactions on Signal Processing* 2005;53(10):3976–87.
- [43] Wang K, Lavoué G, Denis F, Baskurt A, He X. A benchmark for 3D mesh watermarking. In: Proceedings of the IEEE international conference on shape modeling and applications; 2010. p. 231–5.
- [44] Petitcolas FAP, Anderson RJ, Kuhn MG. Attacks on copyright marking systems. In: Proceedings of the international workshop on information hiding; 1998. p. 218–38.
- [45] Wang K, Luo M, Bors AG, Denis F. Blind and robust mesh watermarking using manifold harmonics. In: Proceedings of the IEEE international conference on image processing; 2009. p. 3657–60.
- [46] Vallet B, Lévy B. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum* 2008;27(2):251–60.