DOI 10.1007/s11760-008-0079-y

ORIGINAL PAPER

Watermarking 3D models using spectral mesh compression

Emad E. Abdallah · A. Ben Hamza · Prabir Bhattacharya

Received: 3 December 2007 / Revised: 15 September 2008 / Accepted: 17 September 2008 / Published online: 14 October 2008

Abstract We propose a robust and imperceptible spectral watermarking method for high rate embedding of a watermark into 3D polygonal meshes. Our approach consists of four main steps: (1) the mesh is partitioned into smaller sub-meshes, and then the watermark embedding and extraction algorithms are applied to each sub-mesh, (2) the mesh Laplacian spectral compression is applied to the sub-meshes, (3) the watermark data is distributed over the spectral coefficients of the compressed sub-meshes, (4) the modified spectral coefficients with some other basis functions are used to obtain uncompressed watermarked 3D mesh. The main attractive features of this approach are simplicity, flexibility in data embedding capacity, and fast implementation. Extensive experimental results show the improved performance of the proposed method, and also its robustness against the most common attacks including the geometric transformations, adaptive random noise, mesh smoothing, mesh cropping, and combinations of these attacks.

Keywords 3D watermarking · Mesh compression · Spectral decomposition · Visual error

1 Introduction

The problem of 3D mesh watermarking is a relatively new area as compared to 2D watermarking [1]. It has received less attention partly because the technology that has been used for the image and video analysis cannot be easily adapted to 3D objects that can be represented in several ways including voxels, NURBS, and polygonal meshes. Early algorithms on 3D watermarking [2–4] consist of embedding the watermark information directly by modifying either the 3D mesh geometry or the topology of the triangles. These methods are usually simple and require low computational cost. However, they are not robust enough to different types of attacks. Recently, several watermarking algorithms in the frequency domain have been proposed for 3D mesh [5–7] that are mainly based on multi-resolution mesh analysis (spectral decomposition and wavelet transform) and show good resistance against attacks. In [5] a set of scalar basis functions has been constructed over the mesh vertices where the watermark perturbs the vertices of each mesh along the direction of the surface normal, weighted by the basis functions. In [7] the original mesh is decomposed into a series of details at different scales by using the spherical wavelet transform, and the watermark is then embedded more in the approximation part than in the detail part. In [8] a watermarking scheme for subdivision surfaces has been presented. In [6] a watermarking algorithm based on the mesh spectral matrix has been proposed. The watermark is embedded by modifying the spectral coefficients and this idea was generalized in [9] to watermark point-based 3D geometries. In [10] the normal vector distribution has been used. A blind watermarking scheme robust against affine transformation attacks was proposed in [11]. Watermarking of texture attributes has been proposed in [12]. Two blind watermarking schemes that are robust against distortionless as well as distortion attacks are proposed in [13], where the idea was to modify the vertex norms distribution according to the watermark bit sequence. Wavelet blind watermarking scheme has been proposed in [14] where it is assumed that the host meshes are semi-regular, a wavelet decomposition is applied first to embed the watermark at a suitable resolution level. A robust and fast spectral watermarking scheme for large meshes using a new orthogonal basis functions based on radial basis function has been

E. E. Abdallah (✉) · A. Ben Hamza · P. Bhattacharya
Concordia Institute for Information Systems Engineering,
Concordia University, Montreal, Canada
e-mail: ee_abdal@cs.concordia.ca

proposed in [15]. In [16] the mesh Laplacian matrix was used to encode the 3D shape into a more compact representation by retaining the smallest eigenvalues and associated eigenvectors that contain the highest concentration of the shape information.

Motivated by the need for more robustness against attacks (especially to mesh compression and mesh smoothing), we propose a robust imperceptible watermarking approach using the spectral mesh compression. Our approach uses the mesh Laplacian matrix to embed a watermark in the spectral coefficients of a compressed 3D mesh. Extensive numerical experiments are performed to demonstrate the much improved performance of the proposed method in comparison with other methods. The visual error is evaluated by computing a nonlinear visual error metric between the original 3D model and the watermarked model obtained by our proposed algorithm.

The remainder of this paper is organized as follows. In Sect. 2, we briefly review some background material and describe the spectral compression of the mesh geometry. In Sect. 3 we introduce the proposed approach and describe in detail the watermark embedding and extraction algorithms. In Sect. 4, we present some experimental results and comparisons with existing techniques, and we show the robustness against the most common attacks. Finally, we conclude and point out future directions in Sect. 5.

2 Mesh compression

2.1 3D model representation

In computer graphics and computer-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles. Each edge $e_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$ connects a pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$. Two distinct vertices $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are adjacent (written $\boldsymbol{v}_i \sim \boldsymbol{v}_j$) if they are connected by an edge $e_{ij} \in \mathcal{E}$. The neighborhood of a vertex $\boldsymbol{v}_i$ is the set $\boldsymbol{v}_i^\star = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is the cardinality of $\boldsymbol{v}_i^\star$. Let $\boldsymbol{v}_i = (x_i, y_i, z_i) \in \mathcal{V}$, $1 \le i \le m$, then the mesh vertex matrix $\boldsymbol{V}$ is the $m \times 3$ matrix whose ith row is the vector $\boldsymbol{v}_i$.

2.2 Laplacian matrix of a triangle mesh

The mesh Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{A}$ is the adjacency matrix between the vertices, defined by

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{otherwise} \end{cases}$$

and $\boldsymbol{D}$ is the $m \times m$ diagonal matrix whose (i, i) entry is d_i.

2.3 Spectral mesh compression

In [16] the 3D mesh geometry was represented as a linear combination of a few basis functions. The idea is to apply the eigen-decomposition to the mesh Laplacian matrix, and then discard the largest eigenvalues and their corresponding eigenvectors in order to reduce the dimensionality of the new spectral basis. A significant compression ratio with a very small loss in the mesh quality is obtained because this small number of basis functions contains the optimal concentration of the shape information. The eigen-decomposition of the Laplacian matrix $\boldsymbol{L}$ is given by

$$\boldsymbol{L} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{B}^{\mathrm{T}} \tag{1}$$

where $\boldsymbol{B} = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots\ \boldsymbol{b}_m)$ is an orthogonal matrix whose columns $\boldsymbol{b}_i$ are the eigenvectors of $\boldsymbol{L}$ which we refer to as Laplacian basis functions, and $\boldsymbol{\Lambda}$ =diag$\{\lambda_i : 1 \le i \le m\}$ is a diagonal matrix of the eigenvalues of $\boldsymbol{L}$ arranged in increasing order of magnitude. We express the mesh vertex matrix in the subspace spanned by the Laplacian matrix eigenvectors as follows:

$$\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}} = \sum_{i=1}^{m} \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{b}_i^{\mathrm{T}} \tag{2}$$

where $\boldsymbol{C} = (\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots\ \boldsymbol{c}_m)^{\mathrm{T}}$ is an $m \times 3$ matrix of the spectral coefficient vectors, that is, $\boldsymbol{C} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}$ where $\boldsymbol{C}$ is the projection of the mesh vertex matrix onto the Laplacian basis vectors. Moreover, Eq. (2) can be written as

$$\boldsymbol{V}^{\mathrm{T}} = \underbrace{\sum_{i=1}^{r} \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{b}_i^{\mathrm{T}}}_{\text{compressed}} + \sum_{i=r+1}^{m} \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{b}_i^{\mathrm{T}} = \boldsymbol{C}_r^{\mathrm{T}}\boldsymbol{B}_r^{\mathrm{T}} + \sum_{i=r+1}^{m} \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{b}_i^{\mathrm{T}} \tag{3}$$

where r is usually chosen to be smaller than m, and hence this yields a compressed mesh version $\mathbb{M}_r$ of the original mesh $\mathbb{M}$ with a very small loss in the mesh quality. The matrix $\boldsymbol{B}_r = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots\ \boldsymbol{b}_r)$ contains the spectral basis vectors, and the matrix $\boldsymbol{C}_r = (\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots\ \boldsymbol{c}_r)^{\mathrm{T}}$ contains the spectral coefficient vectors. The spectral coefficients in the x, y, and z-dimension are given by $\boldsymbol{c}_x = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}_x$, $\boldsymbol{c}_y = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}_y$, and $\boldsymbol{c}_z = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}_z$, respectively, see Fig. 1 for an example.

Figure 2 shows two examples of the mesh compression results using Laplacian-based method with 500 basis functions.

2.4 Mesh partitioning

The computation of the eigenvalues and the eigenvectors of a large $m \times m$ Laplacian matrix is prohibitively expensive $\mathcal{O}(m^3)$. To circumvent this limitation, we partition a large 3D mesh into smaller sub-meshes. The embedding and extraction algorithms are then applied to each sub-mesh. In our

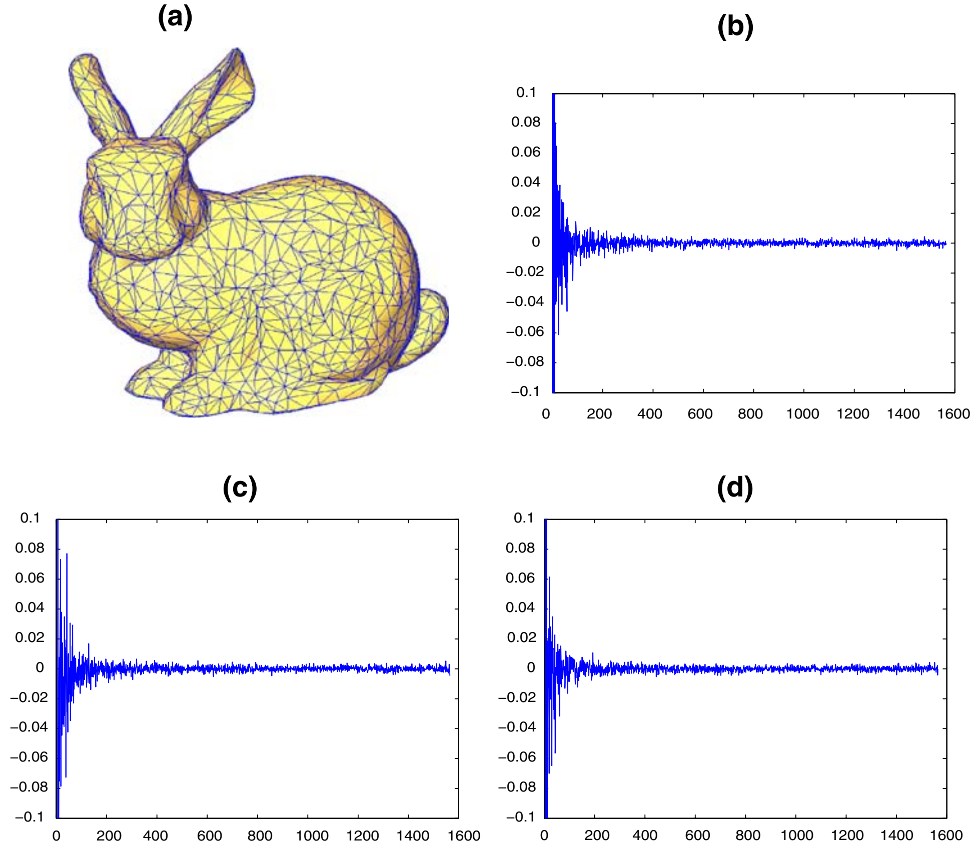
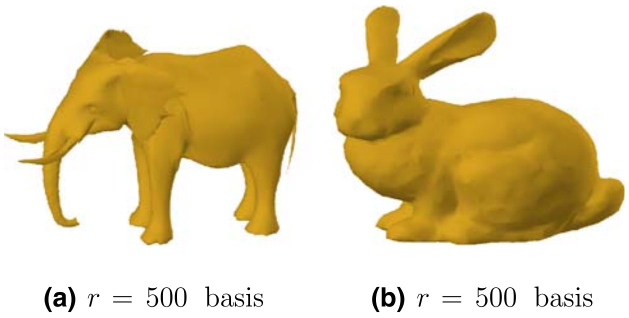
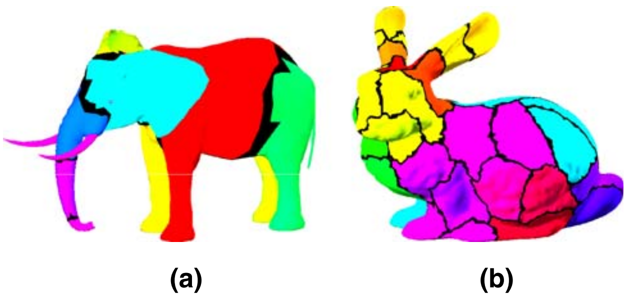


Fig. 1 **a** 3D bunny model, and its spectral coefficients in the **b** x-dimension, **c** y-dimension, and **d** z-dimension



(a) $r = 500$ basis **(b)** $r = 500$ basis

Fig. 2 Spectral compression of the 3D models using Laplacian compression. **a** Elephant model, **b** Bunny model



(a) (b)

Fig. 3 MeTis mesh partitioning. Each sub-mesh is colored by a random color. *Black triangles* represent edge cuts. **a** Elephant model: 4,067 vertices, 8 sub-meshes. **b** Bunny model: 20,100 vertices, 40 sub-meshes

approach we used the MeTiS software [17] for mesh partitioning, and we used sub-meshes of 500 vertices on average as illustrated in Fig. 3.

2.5 Watermarking in the mesh spectral domain

Watermarking schemes in the mesh spectral domain usually embed the watermark data into the mesh shape by modifying the spectral coefficients computed from the mesh topology. It is known that the smaller spectral coefficients correspond to the low frequency components, and the high spectral coefficients correspond to the high frequency components of the 3D mesh.

The two methods proposed in [6,21] use a spectral approach by embedding the watermark directly in the spectral coefficients computed from the original 3D mesh.

Our experiments show that in order to increase the robustness against smoothing and compression attacks, the watermark should be embedded only in the spectral coefficients that represent the low frequency components. However, in order to increase the robustness against additive random noise the same watermark should be embedded repeatedly as much as possible in the spectral coefficients vectors. So there is a trade-off between the robustness against random noise attack on the one hand and smoothing and compression attacks on the other hand.

In [6], the lowest five spectral vectors (low frequency components) are not used in the embedding process due to their use in the realignment process before the extraction algorithm. In [21], the realignment algorithm is fixed, and the lowest frequency components are used to embed the watermark. However, the trade-off between smoothing, compression and noise attacks was not addressed.

Motivated by the need for more robustness against attacks we use the "Spectral compression of mesh geometry" introduced in [16]. We also use the fact that the low frequency coefficients represent the global shape features of a 3D mesh (the rough approximation of the model may be reconstructed using small low frequency spectral coefficients). The goal of our proposed scheme is to embed the watermark in the low frequency components by repeating the watermark embedding process as much as possible. The number of the spectral coefficients of the compressed 3D mesh is exactly the same as the number of the spectral coefficients of the original 3D mesh without compression. As a result, we obtain the maximum number of the watermark repletion (maximum robustness against a noise attack). Moreover, all the embedded watermarks are done in the low frequency components that are used during the mesh compression stage. This guarantees the most robustness against smoothing and compression attacks.

3 Proposed method

In this section, we describe the main steps of the proposed watermark embedding and extraction algorithms, Figs. 4 and 5 show the flow diagrams. The goal of our proposed approach may be described as embedding the watermark in the global shape features which are represented by the low frequency components of the 3D mesh. In this case we are not only increasing the robustness against attacks but also increasing the watermark imperceptibility. The proposed algorithm embeds the watermark information into the spectral coefficients of the compact representation of the 3D model.

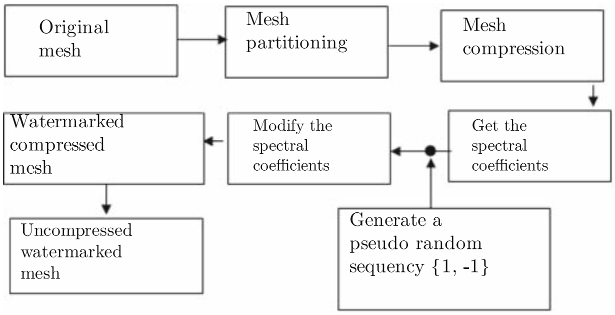

Original mesh
Mesh partitioning
Mesh compression
Watermarked compressed mesh
Modify the spectral coefficients
Get the spectral coefficients
Uncompressed watermarked mesh
Generate a pseudo random sequency {1, -1}

Fig. 4 Watermark embedding process

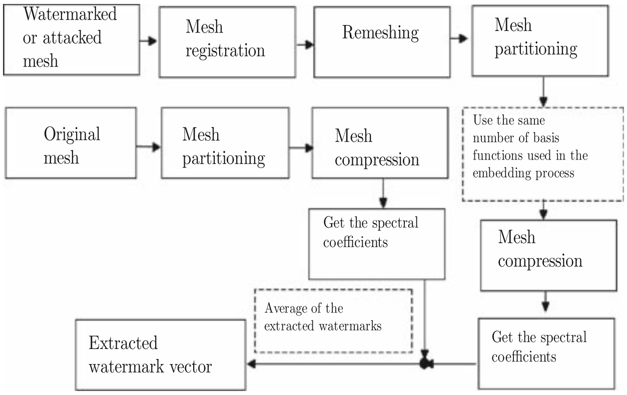

Watermarked or attacked mesh
Mesh registration
Remeshing
Mesh partitioning
Original mesh
Mesh partitioning
Mesh compression
Use the same number of basis functions used in the embedding process
Get the spectral coefficients
Mesh compression
Average of the extracted watermarks
Extracted watermark vector
Get the spectral coefficients

Fig. 5 Watermark extraction process

3.1 Watermark embedding process

The 3D mesh is partitioned into smaller sub-meshes and the watermark embedding procedure is applied to each sub-mesh. Let $\mathbb{S}$ be a sub-mesh of n vertices and W be a pseudo-random vector of $\{-1,1\}$ used as a watermark of size k such that $k \ll n$. For all sub-meshes the watermark embedding process consists of the following steps:

1. Compute the Laplacian matrix $\boldsymbol{L}$ of size $n \times n$.
2. Compute the eigenvalues and the associated eigenvectors (basis functions) of $\boldsymbol{L}$.
3. Project the mesh vertices onto the basis functions to get the spectral coefficients matrix $\boldsymbol{C} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}$ of the original sub-mesh $\mathbb{S}$.
4. Use the r basis functions ($r < n$) to obtain the compressed sub-mesh $\mathbb{S}_r$.
5. Repeat the steps (1–3) on the compressed 3D sub-mesh $\mathbb{S}_r$ to get the spectral coefficients matrix $\boldsymbol{C}_r$.
6. Duplicate the watermark d times, where $d = \lfloor n/k \rfloor$. Let the new watermark sequence be W_d. Modify the compressed spectral coefficients $\boldsymbol{C}_r$ by the watermark sequence W_d. So,

$$\widehat{\boldsymbol{C}}_r = \boldsymbol{C}_r + \alpha W_d \tag{4}$$

where $\widehat{\boldsymbol{C}}_r$ is the modified compressed spectral coefficients matrix, and α is a constant (watermark strength).

7. Express the compressed watermarked sub-mesh vertices in the subspace using the modified spectral coefficients.

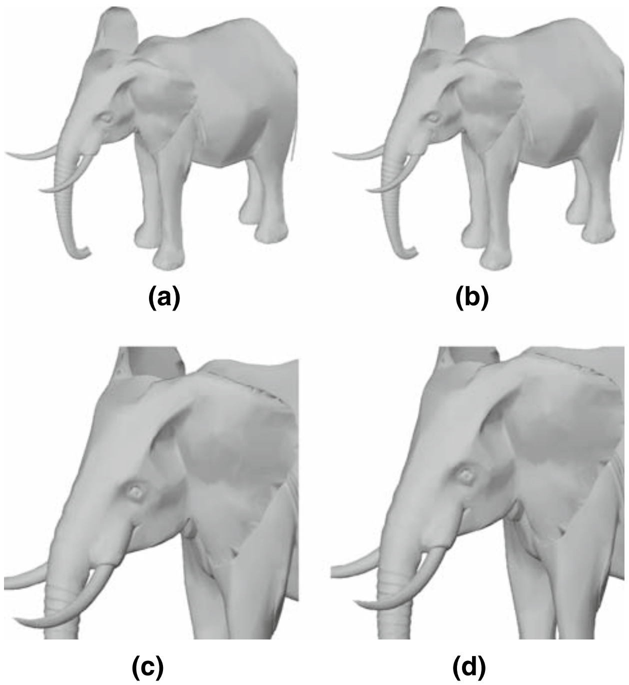


Fig. 6 **a**, **b** Original 3D model with the corresponding watermark model. **c**, **d** Close-up comparison. Elephant model with (4,076 vertices, 7,999 faces)

Thus,

$$\boldsymbol{V}_{W_r}^{\mathrm{T}} = \widehat{\boldsymbol{C}}_r^{\mathrm{T}} \boldsymbol{B}^{\mathrm{T}} = \sum_{i=1}^{r} \widehat{\boldsymbol{c}}_{i_r}^{\mathrm{T}} \boldsymbol{b}_i^{\mathrm{T}}$$

where $\boldsymbol{V}_{W_r}^{\mathrm{T}}$ is the compressed sub-mesh vertex matrix.

8. Use the remaining basis functions that are not used in step (4) to obtain the uncompressed watermarked sub-mesh with vertex matrix given by

$$\boldsymbol{V}_{W}^{\mathrm{T}} = \boldsymbol{V}_{W_r}^{\mathrm{T}} + \sum_{i=r+1}^{n} \boldsymbol{c}_i^{\mathrm{T}} \boldsymbol{b}_i^{\mathrm{T}} \tag{5}$$

where $\boldsymbol{C} = \{\boldsymbol{c_i}\}_{r+1}^{n}$ is the spectral coefficients matrix of the high frequency basis functions. Figure 6a, b shows a 3D model elephant with the corresponding watermarked model. In Fig. 6c, d we used the zoom tool enlarge the view of the 3D Elephant models head in order to clearly show the performance of our proposed algorithm.

3.2 Watermark extraction process

We present a private watermarking scheme, meaning that the original unwatermarked object is needed for the extraction process. The original model helps getting a perfect realignment with the watermarked model before starting the extraction process. In addition to the original model, the embedding strength factor, basis number and the sizes of mesh-partitions need to be saved in the secret key. An initial search step to find the right original model from the owner database is needed. We used the global geodesic measure proposed in [18]. The idea is to represent an object by a probabilistic shape descriptor that measures the global geodesic distance between two arbitrary points on the surface of an object. Unlike the Euclidean distance which is more suitable for linear spaces, the geodesic distance has the advantage of being able to capture the intrinsic geometric structure of the data. The matching task therefore becomes a one-dimensional comparison problem between probability distributions which is clearly much simpler than comparing 3D structures. The computational complexity of this method is $\mathcal{O}(m \log m)$, where m is the number of the centroids points of a 3D mesh model. Let the original unwatermarked mesh be $\mathbb{M}$ and the watermarked probably attacked mesh be $\widehat{\mathbb{M}}$.

(i) Mesh Registration

We need to estimate the optimal rotation, scaling and translation to get $\widehat{\mathbb{M}}$ back to its initial scale and location if it is changed. This registration process is very important in order to extract the watermark successfully. We use the iterative closest point (ICP) method [19,20] to select the optimal transformation (translation and rotation) to align two surfaces. Sometimes it is necessary to provide initial alignment, especially with the cropping attack. For the scaling transform, if both meshes represent the non-cropped objects or represent exactly the same surface patches of an object, we need to align both meshes to their initial position using ICP and then measure the ratio between the length of their corresponding axes.

(ii) Remeshing

After registration, a remeshing is usually necessary to deal with the changes resulted by the attacks that may modify the mesh topology like simplification algorithms. To map the original topology we used the remeshing method [21] by tracing a ray through each vertex of the original mesh in the same direction of the normal vector of that vertex. If an intersection point is not found, create a vertex with the same coordinate as its reference in the original mesh. After applying the registration and remeshing processes to the watermarked and probably attacked mesh, we apply the watermark extraction algorithm which can be summarized as follows: The 3D mesh is partitioned into smaller sub-meshes using the same procedure as in the embedding process. For each sub-mesh:

1. Apply the first four steps of the embedding process with the same number of basis functions to the initial and the watermarked sub-meshes to obtain compressed version.

2. Apply the steps (1–3) of the watermarking algorithm on the compressed 3D sub-meshes. Then, to extract the watermark vector we compare the spectral coefficients of the initial compressed sub-mesh with the spectral coefficients of the watermarked and probably attacked compressed sub-mesh. $w_x^i = (\widehat{x_i} - x_i)/\alpha$, $w_y^i = (\widehat{y_i} - y_i)/\alpha$, and $w_z^i = (\widehat{z_i} - z_i)/\alpha$, where $(\widehat{X}, \widehat{Y}, \widehat{Z})$, (X, Y, Z) are the spectral coefficients of the compressed watermarked and the compressed initial sub-meshes, respectively, and α is a constant saved in the secret key during the embedding process.
3. Construct $\overline{W} = (W_x + W_y + W_z)/3$, where W_x, W_y, and W_z are the extracted watermark vectors in step (2).
4. Find the average watermark vector $\overline{W}_d$ from $\overline{W}$ which contains $d = \lfloor n/k \rfloor$ watermark copies. Finally the extracted vector is given by the decision rule:

$$\widehat{W}_d = \{\overline{w}_{d_i}\}_{i=1}^{k} = \begin{cases} -1 & \text{if } w_{d_i} < 0 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

5. If the correlation coefficient between $\widehat{W}_d$ and W is greater than a predefined threshold, then the watermark is present.

4 Experimental results

Our experiments were performed using a variety of 3D models represented as triangle meshes. Table 1 shows the characteristics of the 3D models used in our experiments (collected by courtesies of the Stanford University, Avalon, Cyberware, and the Max Planck Institute). We conducted experiments to test the imperceptibility of the watermark and the robustness against attacks. A comparison between our proposed scheme and other spectral watermarking techniques were also conducted.

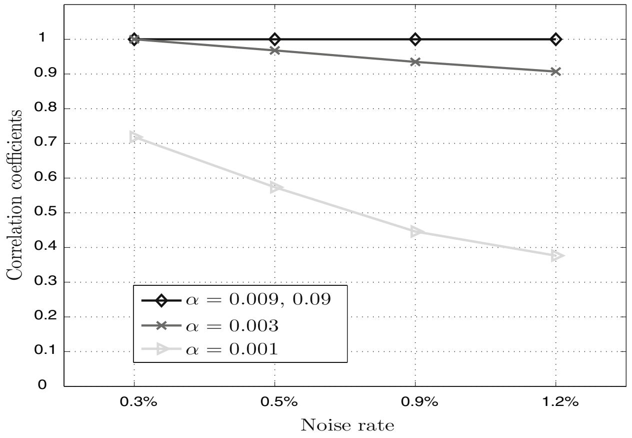


Fig. 7 Correlation coefficient results for the camel model using four different strength factors and four noise rates attacks

4.1 Imperceptibility

In order to achieve high visual quality of the watermarked model, the watermark strength factor α should be taken into consideration. The most common embedding rule is the additive one: $\overline{x}_i = x_i + \alpha w_i$ where x_i is the ith component of the original vector, w_i the ith sample of the watermark, and α is the watermark strength. The parameter α is chosen by the owner of the 3D model such that it is small enough to keep the watermark imperceptible to the human observer, and large enough to resist as many attacks as possible. Figures 7 and 8 depict the robustness of the camel model with noise and smooth attacks, respectively. Different strength factors, different noise rates, and different smoothing iterations have been used. Clearly, a higher strength factor gives better correlation coefficients between the original watermark vector and the average vector of the extracted watermarks.

Figure 9 shows an example of the influence of the strength factor on the watermark perceptibility. The watermark embedded in the 3D models with a high strength factors is not only perceptible to the human observer but also it may destroy

Table 1 characteristics of the 3D models used in our experiments

Model	No. of vertices	No. of faces	No. of patches	No. of watermarks
Camel	4,001	8,050	7	750
Bunny	20,100	39,999	40	3,768
Max Plank	5,040	10,067	10	945
Elephant	4,067	7,999	8	762
Tank	15,186	13,902	31	2,847
Mesh part	2,496	5000	4	468
Cow	2,903	5,804	4	543
Rocker arm	10,000	20,000	18	1,875
Hand	10,113	19,801	18	1,896

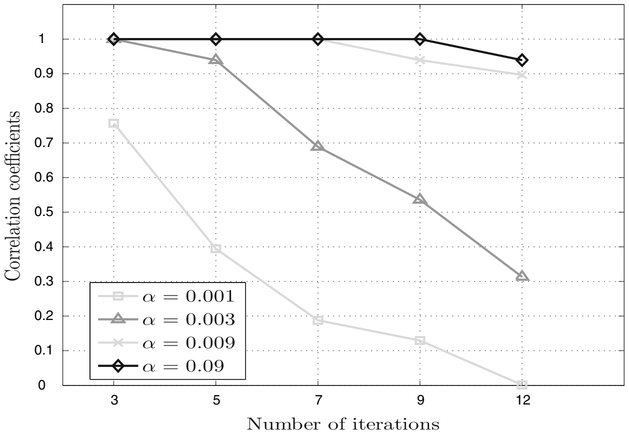
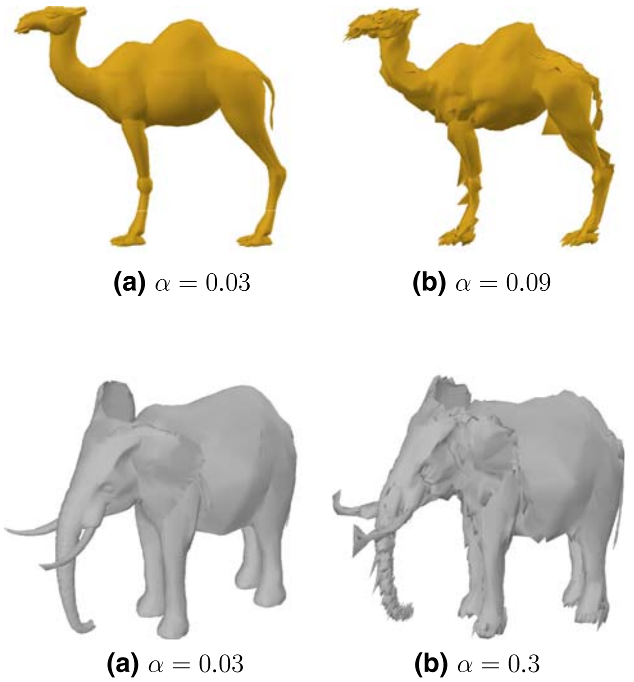


Fig. 8 Correlation coefficient results for the camel model using four different strength factors and smoothing attacks with different number of iterations


(a) $\alpha = 0.03$ **(b)** $\alpha = 0.09$

(a) $\alpha = 0.03$ **(b)** $\alpha = 0.3$

Fig. 9 Watermark perceptibility. The *watermarks* embedded in the Camel and Elephant models with different strength factors

the overall geometric structure of the models. So there is a trade-off between the robustness of the watermarked model against attacks and the degradation of the original 3D mesh. Figure 10 shows the best strength factor α for the 3D camel model and it is chosen based on the minimization of the error rate.

To quantify the imperceptibility of the proposed approach, we propose a nonlinear visual error $D(\mathbb{M}, \widehat{\mathbb{M}})$ defined between the original model $\mathbb{M}$ and the watermarked model $\widehat{\mathbb{M}}$ as follows

$$D(\mathbb{M},\widehat{\mathbb{M}})=\left(\sum_{i=1}^{m}\|\boldsymbol{v}_i-\hat{\boldsymbol{v}}_i\|^2+\|\mathcal{A}(\boldsymbol{v}_i)-\mathcal{A}(\hat{\boldsymbol{v}}_i)\|^2\right)\Big/(2m)$$

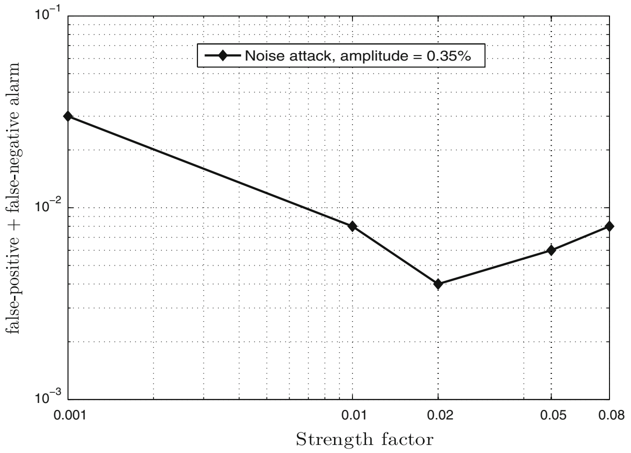


Fig. 10 False-positive and false-negative alarm for different values of the strength factor α

where $\{\boldsymbol{v}_i\}_{i=1}^{m}$ and $\{\hat{\boldsymbol{v}}_i\}_{i=1}^{m}$ are the mesh vertex sets of $\mathbb{M}$ and $\widehat{\mathbb{M}}$, respectively. $\mathcal{A}$ is a nonlinear diffusion operator [22] defined as

$$\mathcal{A}(\boldsymbol{v}_i)=(1/d_i)\sum_{\boldsymbol{v}_j\in\boldsymbol{v}_i^{\star}}(\boldsymbol{v}_i-\boldsymbol{v}_j)\Big(g(|\nabla\boldsymbol{v}_i|)+g(|\nabla\boldsymbol{v}_j|)\Big) \quad (7)$$

where the gradient magnitudes are given by

$$|\nabla\boldsymbol{v}_i|=\sqrt{\sum_{\boldsymbol{v}_j\in\boldsymbol{v}_i^{\star}}\left\|(\boldsymbol{v}_i/\sqrt{d_i})-(\boldsymbol{v}_j/\sqrt{d_j})\right\|^2}, \quad (8)$$

$$|\nabla\boldsymbol{v}_j|=\sqrt{\sum_{\boldsymbol{v}_k\in\boldsymbol{v}_j^{\star}}\left\|(\boldsymbol{v}_j/\sqrt{d_j})-(\boldsymbol{v}_k)/(\sqrt{d_k})\right\|^2} \quad (9)$$

and $g(x) = 1/(1 + x^2/c^2)$ is the Cauchy weight function (see Fig. 11) with a constant tuning parameter c that needs to be estimated. It can be shown (see [23]) that the 95% asymptotic efficiency on the standard Gaussian distribution is obtained with $c = 2.3849$ which is used in all the experimental results. Note that the visual error $D(\mathbb{M}, \widehat{\mathbb{M}})$ requires the use of two neighboring rings as depicted in Fig. 12. Intuitively, the anisotropic operator $\mathcal{A}$ introduces some smoothing effect which may be explained as follows: around the sharp features of the 3D mesh where the vertex gradient magnitudes are large, the non-linear diffusion operator in Eq. (7) used to preserve the sharp feature of the 3D mesh. Moreover, in the flat regions of the 3D mesh where the vertex gradient magnitudes are relatively small, Eq. (7) is reduced to a linear operator which tends to checks the distortion of the watermarked model in these flat areas (smoothness of the watermarked model).

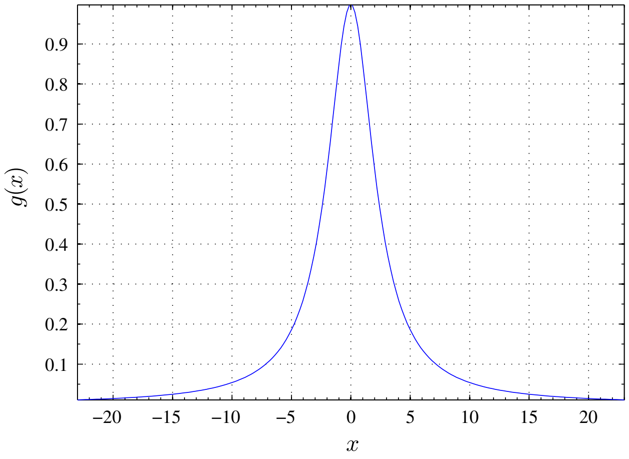


Fig. 11 Cauchy weight function with $c = 2.3849$

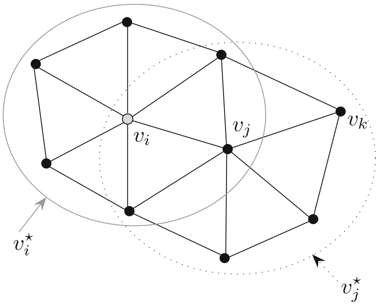


Fig. 12 Illustration of two neighboring rings

We used also the geometric Laplacian distance error [16,24] defined as

$$G(\mathbb{M}, \widehat{\mathbb{M}}) = \left(\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\| + \|\mathcal{I}(\boldsymbol{v}_i) - \mathcal{I}(\hat{\boldsymbol{v}}_i)\|^2 \right) \Big/ (2m)$$

where $\mathcal{I}$ is the geometric Laplacian operator given by

$$\mathcal{I}(\boldsymbol{v_i}) = \boldsymbol{v}_i - \left(\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^*} \ell_{ij}^{-1} \boldsymbol{v}_j \right) \Big/ \left(\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^*} \ell_{ij}^{-1} \right) \tag{10}$$

where ℓ_{ij} is the Euclidian distance between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. This visual metric consists of a sum of two error terms between the original and the watermarked vertex positions: the first term provides a measure of geometric closeness between the correct and the watermarked vertex locations, and the second term captures the smoothness properties of these vertices.

The proposed metric error is a nonlinear extension of the geometric distance error introduced in [16]. The main difference between the two metric errors is that the second term of geometric distance error is defined in terms of a linear operator that tends to smooth more, whereas the second term of the non-linear diffusion operator that tends to smooth less and hence leads to a much better performance of the mesh geometric structures. Figure 13 shows the nonlinear diffusion operator error and the geometric Laplacian distance error for three different watermarked models with different strength factors. Clearly our experimental results show that the proposed method gives low visual metric errors that guarantee the imperceptibility of the watermark.

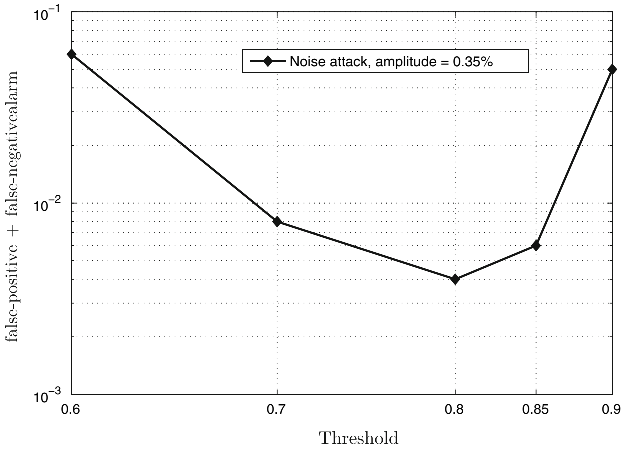


Fig. 14 False-positive and false-negative alarm for different values of the threshold

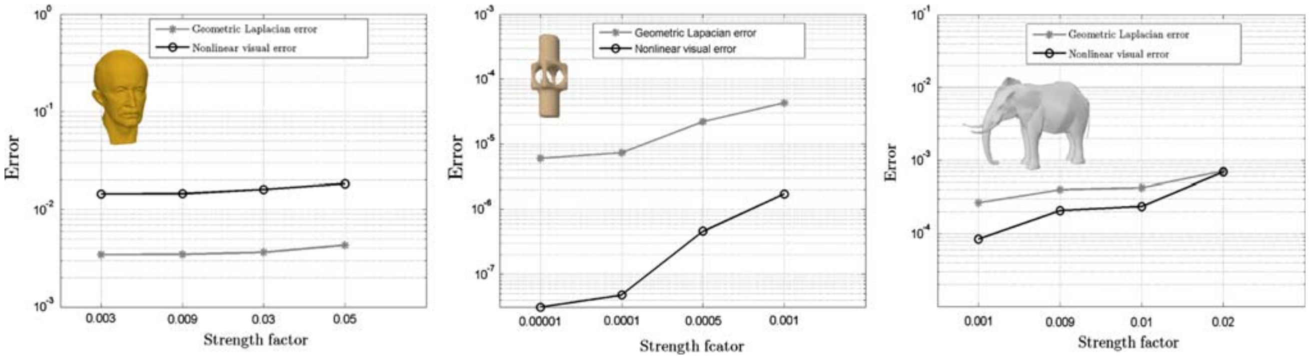
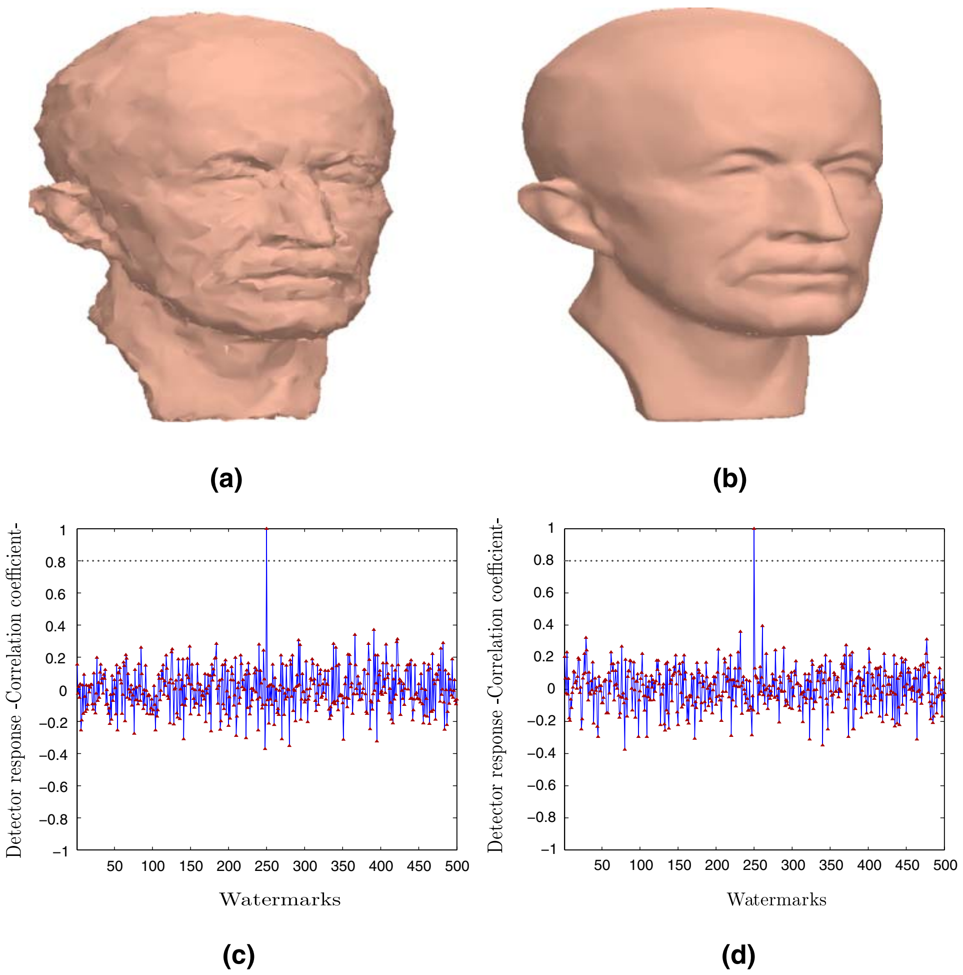


Fig. 13 Nonlinear visual error and the geometric Laplacian distance error changes with different strength factors **a** Max Planck, **b** mesh part, and **c** elephant

Fig. 15 Robustness against random noise and Laplacian smoothing attacks. **a** Max Planck model with ($\sigma^2 = 0.0035$) additive noise. **b** Max Planck model after (7 iterations) of the low pass filter. **c**, **d** Detector responses for **a** and **b**, respectively



4.2 Robustness

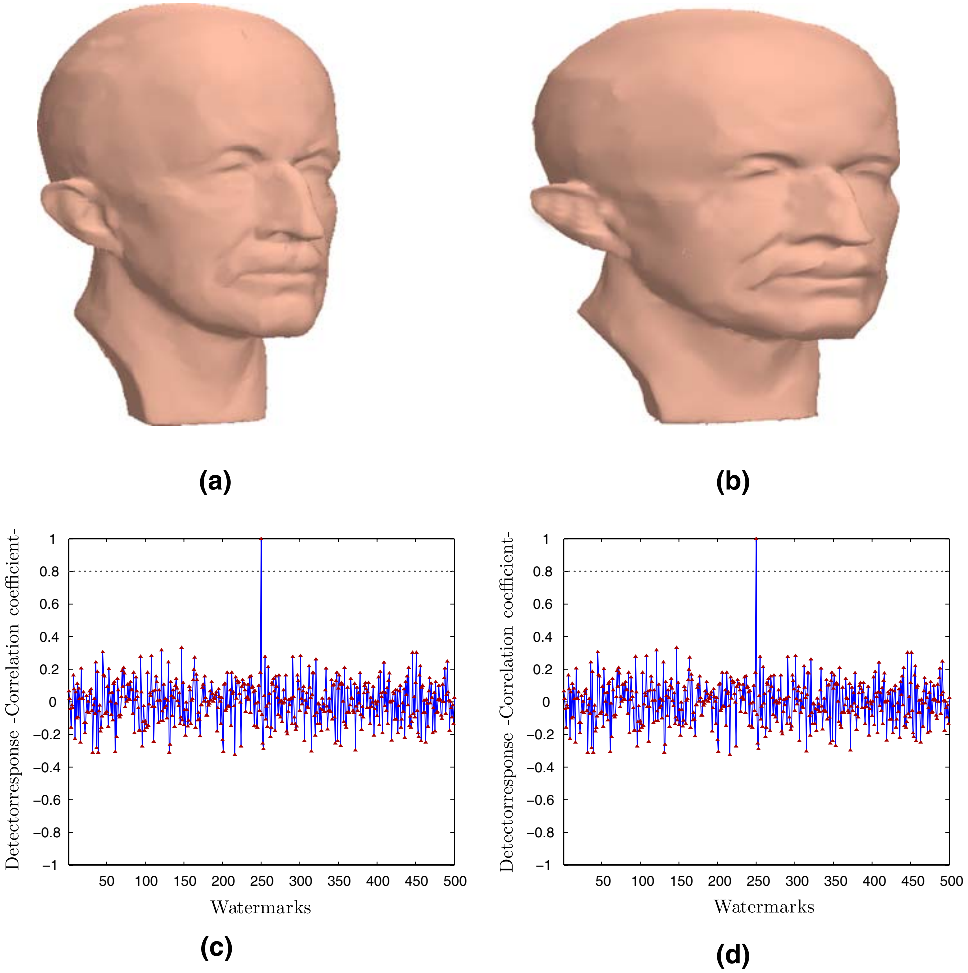
Robustness is an important factor that we need to consider when designing a watermark system for copyright protection. Attacks do not necessarily mean the removal of the watermark; they can be operations to make the watermark undetectable [25,26]. We tested the robustness of the proposed algorithm with different 3D models (see Table 1) against various attacks including mesh transformation, mesh simplification, additive random noise, mesh smoothing, compression, and cropping. Sequences of 32 binary digits $\{-1, 1\}$ are randomly generated and used as watermarks. In the experiments we display the attacked models with the detector response for the real watermark, and 499 randomly other generated watermarks. For all the detector response figures the correlation between the original watermark and the extracted watermark is located at 250 on the X-axis and the dotted line at 0.8 on the Y-axis represents the threshold. The threshold is chosen to decrease false-positive (presenting incorrectly the watermark in the model) and false-negative alarm (falling to detect the watermarked model). After running our algorithm 500 times for the noise attack (amplitude 0.35%), we calculated the false-positive and false-negative alarm for different values of the threshold. The algorithm run the best as depicted in Fig. 14 with threshold = 0.8. If the correlation is larger than 0.8, then the watermark is present.

The non-linear diffusion operator is used not only to quantify the imperceptibility of the proposed approach, but also to estimate the watermarking strength factor leading to a much better performance of the mesh geometric structures. More specifically, the non-linear diffusion measure is calculated for every sub-mesh by using a slightly lower strength factor for all the sub-meshes with a high value of the diffusion operator and a higher strength factor for all the sub-meshes with a low value of the diffusion operator. This adaptive strength factor leads to an increased improvement of the robustness against attack.

4.2.1 Additive random noise

In order to test the robustness of the watermark, an additive Gaussian noise was added to the watermarked mesh by

Fig. 16 Robustness against geometric transformation and compression attacks. **a** Max Planck model is scaled in Z direction by factor of 2 then rotated by 20° around Y-axis. **b** Compressed Max Planck model of 3,000 basis functions. **c**, **d** Detector responses for **a** and **b**, respectively



summing a random vector to each vertex in the model. See Fig. 15a for the attacked Max Planck model by Gaussian random noise ($\sigma^2 = 0.0035$). The watermark could be extracted without any loss. The detector response is illustrated in Fig. 15c. The watermark is lost when we increased the noise ($\sigma^2 = 0.0045$) for the Max Planck model.

4.2.2 Mesh smoothing

Smoothing algorithms may be used by an attacker to destroy the watermark by moving the node geometry of the watermarked mesh. We used the Laplacian filter algorithm [27] that adjusts the location of each mesh vertex to the centroid of its neighbouring vertices. Hence the high frequency components are those that are most affected by low pass filtering. Our proposed algorithm is robust against smoothing attack as we expected because the watermark was embedded in the low frequency components. Figure 15b depicts the attacked Max Planck model by seven smoothing iterations, and Fig. 15d shows the detector response. As can be seen, the mesh is significantly smoothed but the watermark is still perfectly detectable.

4.2.3 Geometric transformations

These are the simplest attacks used to test the watermark detectors. Like other watermarking algorithms that use the 3D registration and the original model, the proposed approach is robust against geometric attacks because the transformations that are applied to the mesh can be inverted using mesh registration. In Fig. 16a the attacked Max Planck model is obtained in two steps. First, the model is scaled in the Z direction by a factor of 2. Second, the scaled model is rotated around Y-axis by 20°. Figure 16c depicts the watermark extraction response after the registration process was applied. Clearly the detector is still able to recover the watermark.

4.2.4 Mesh compression

Mesh compression has recently become one of the most effective attacks because the new compression techniques

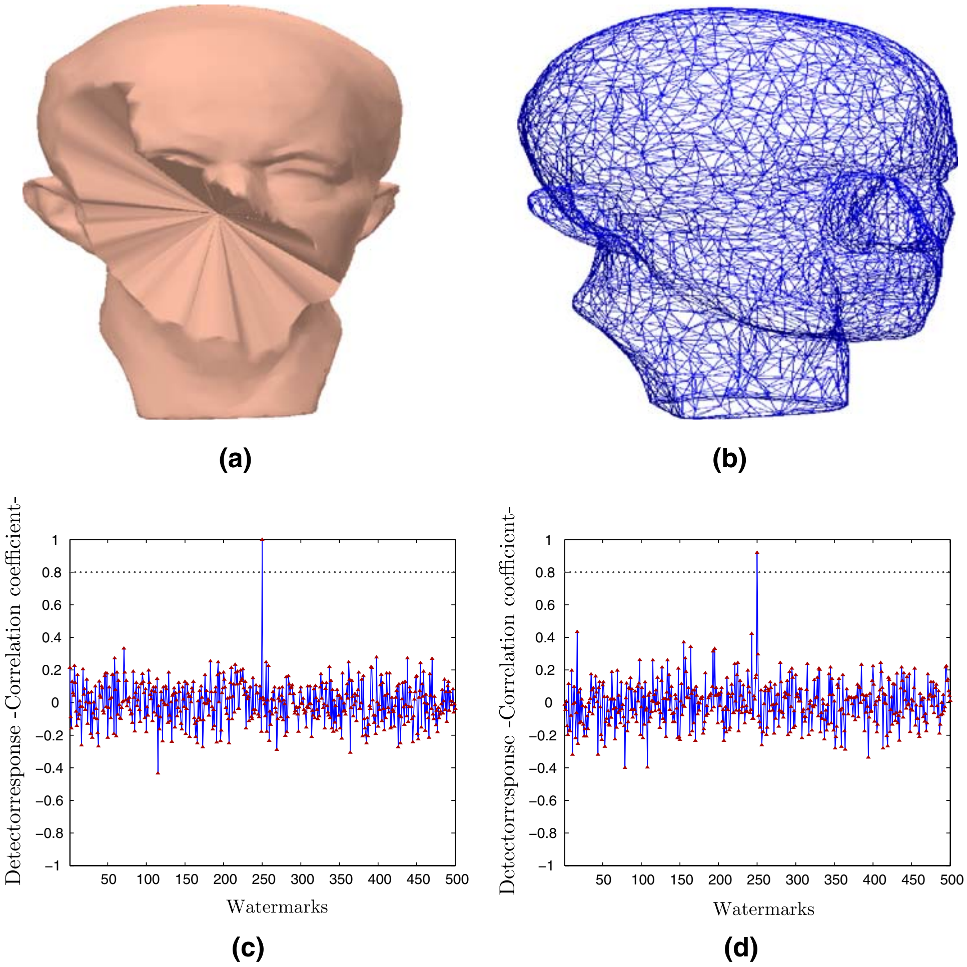


Fig. 17 Robustness against cropping and mesh simplification attacks. **a** Cropped 600 vertices from Max Planck model. **b** Max Planck simplified down to 2,502 vertices and 5,000 faces. **c**, **d** Detector responses for **a** and **b**, respectively

[16,28,29] reach a very significant compression ratio with very small loss in the mesh quality. We evaluated the robustness of our method against a compression attack [16]. The proposed method is robust against compression because the watermark is embedded in the spectral coefficient of the compressed mesh. Figure 16b depicts the compressed Max Planck model constructed with 3,000 basis functions from the original mesh of 5,040 basis functions. The detector response is shown in Fig. 16d.

4.2.5 Mesh cropping

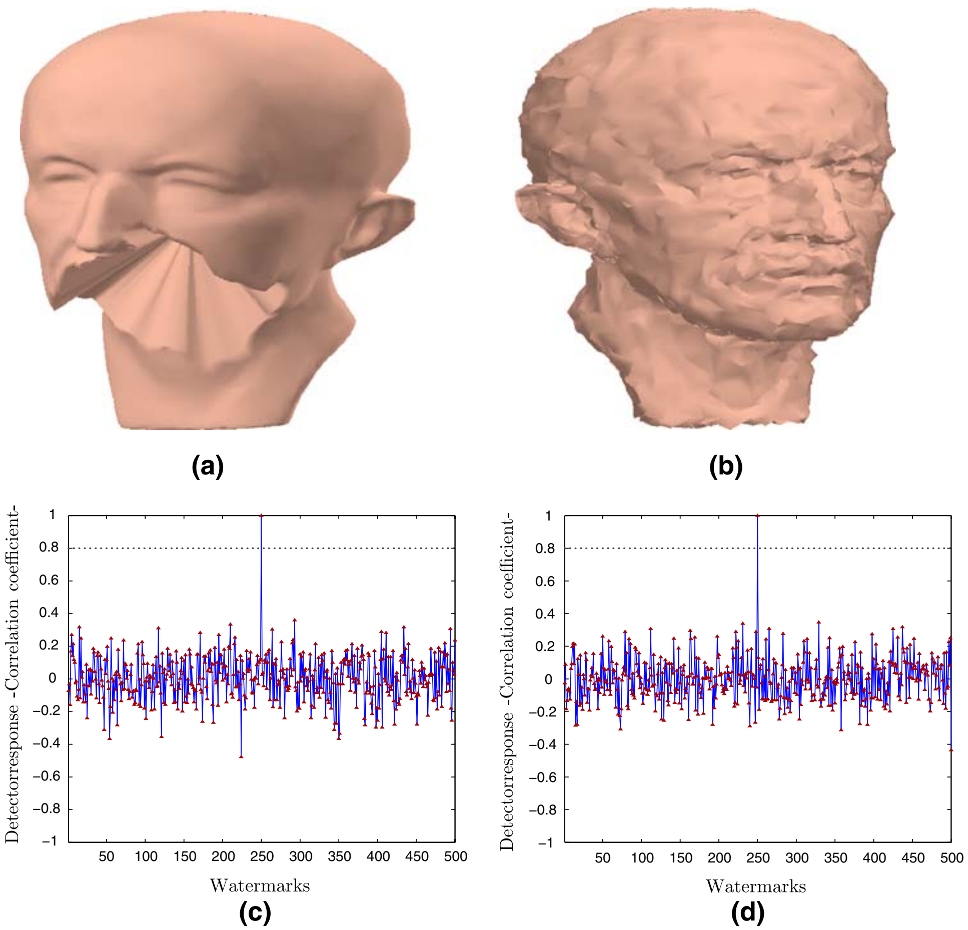
This technique may be used by an attacker to destroy the watermark by removing part of the watermarked mesh. We verified the robustness of the proposed scheme against mesh cropping by trying to extract the watermark from the cropped 3D mesh. Since the watermark is embedded repeatedly using mesh partitioning the watermark can be fully recovered from the deteriorated cropped mesh. Figure 17a depicts the cropped Max Planck model (600 vertices have been removed). The watermark is recovered perfectly from the cropped model as it is shown in the detector response in Fig. 17c.

4.2.6 Mesh simplification

This method may also be used by an attacker to reduce the number of faces of the 3D mesh. This reduction could remove or destroy the watermark. See Fig. 17b for the simplified Max Planck model. The mesh is simplified down from 5,040 vertices and 10,067 faces to 2,502 vertices and 5,000 faces. Our proposed method is robust against the simplification attack because of the remeshing process. The detector response for the attacked mesh in Fig. 17b is illustrated in Fig. 17d.

We also tested the performance of our proposed algorithm using a combination of the previous attacks. Figure 18a, b show the watermarked Max Planck model with multiple attacks. In Fig. 18a the watermarked model is passed through low pass filtering (7 iterations), and then a cropping attack has been applied to remove 540 vertices from the smoothed

Fig. 18 Robustness against multiple attacks. **a** Max Planck model attacked with smoothing (8 iterations) and cropping 540 vertices. **b** Max Planck model attacked with additive noise ($(\sigma^2 = 0.0025)$) then simplified to (80%) of original faces. **c**, **d** Detector responses for **a** and **b**, respectively



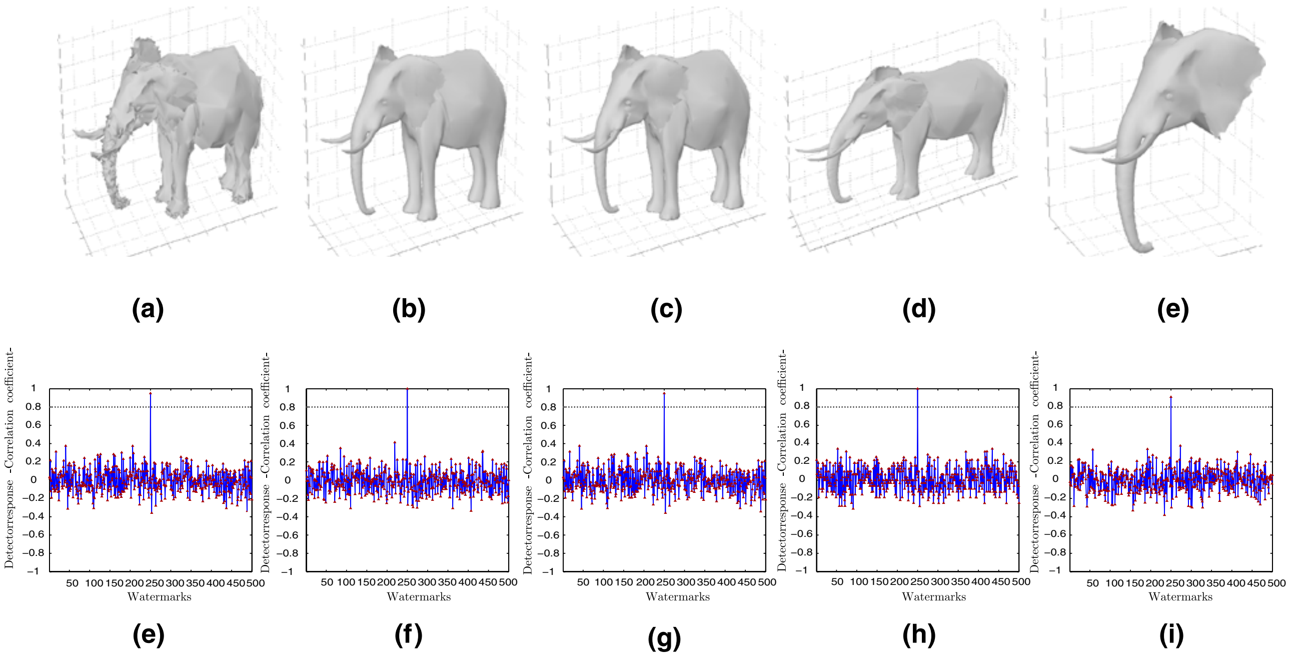


Fig. 19 Watermarked elephant model with different attacks and their corresponding detector responses: for each attack the correlation coefficient between the extracted watermark and 499 different random watermarks are shown (250 on X-axis is the correlation with the real watermark). **a**, **e** Additive noise ($\sigma^2 = 0.007$), **b**, **f** Low pass filter 9 iterations, **c**, **g** Compression 1,500 basis functions, **d**, **h** scaling in X direction, **e**, **i** cropping 1700 vertices and smoothing seven iterations

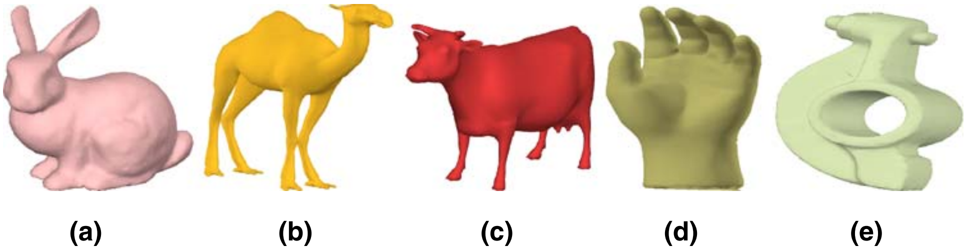


Fig. 20 Smoothing attack with different iteration numbers: **a** 12, **b** 13, **c** 10, **d** 30 and **e** 12

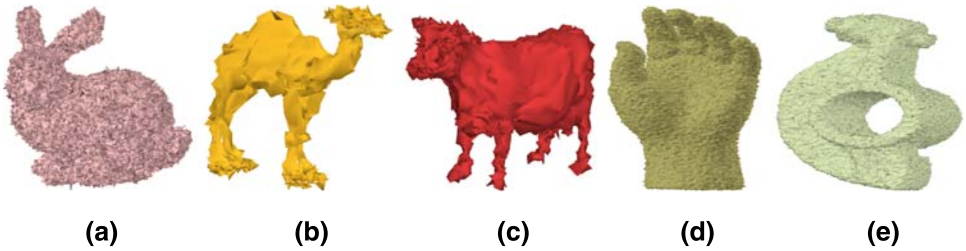


Fig. 21 Gaussian noise attack with different standard deviations: **a** 0.013, **b** 0.013, **c** 0.0095, **d** 0.0095 and **e** 0.0095

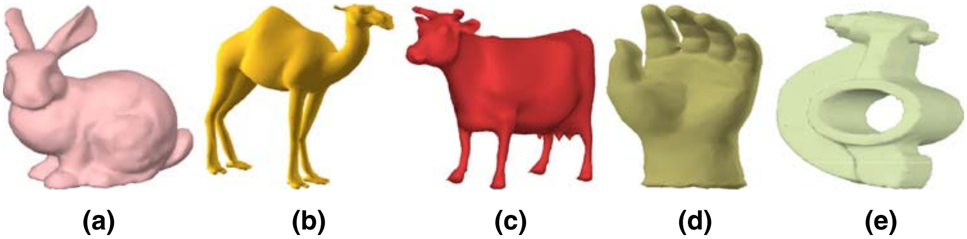


Fig. 22 Compression attack with different numbers of basis functions: **a** 800, **b** 250, **c** 400, **d** 1,000 and **e** 600

mesh. Figure 18b depicts the attacked model after adding additive random noise of ($\sigma^2 = 0.0025$) and being simplified down to 80% of the original vertices. In both cases the proposed algorithm was able to recover the watermark fully [see the detector responses for (a,b) in (c,d), respectively]. More experiments with different models are shown in Fig. 19.

4.3 Comparisons with existing techniques

We conducted several experiments to compare the robustness of the proposed method with related existing techniques that use mesh spectral coefficients vectors to embed the watermark in the frequency domain, and in particular with watermarking 3D meshes in the spectral domain [6] and its extension [21]. In our experiments, we embedded the watermark in all the spectral coefficients vectors. However, in [6] the lowest five spectral coefficients are used to realign the original and the watermarked mesh before applying the watermark extraction process. So in [6] the watermark is embedded in all the $(n - 5)$ higher spectral coefficients.

In our experimental comparisons we used six different 3D models: camel, elephant, bunny, cow, rocker-arm, and hand, and a sequences of 16 binary digits $\{-1, 1\}$ are randomly generated and used as watermarks. For each attack we used various strengths. Table 2 shows the comparison results of the proposed watermarking scheme with the methods introduced in [6,21] against smoothing attack. Three different

Table 2 Comparison results: robustness against smoothing attack

Model	α	No. of iterations	Corr. proposed	Corr. [21]	Corr. [6]
Camel	0.02	11	**1**	0.6831	0.5164
		13	**1**	−0.5222	−0.4229
		15	**0.8783**	−0.9853	−0.6181
Elephant	0.02	10	**1**	0.5146	0.2
		12	**0. 8783**	0.0667	−0.2437
		15	**0.8783**	−0.3333	−0.3333
Bunny	0.0002	10	**1**	0.6	0.4667
		12	**1**	0.3333	−0.4472
		15	**1**	−0.2	−0.5164
Cow	0.01	10	**1**	0.6831	0.3333
		15	**0.8783**	0	0.0667
		18	**0.7746**	−0.5164	−0.7746
Rocker-arm	0.0001	10	**1**	0.8783	0.8783
		13	**1**	−0.5164	−0.7333
		15	**0.8783**	−0.7333	−0.8704
Hand	0.09	35	**1**	0.8704	0.7454
		40	**1**	0.7746	0.6202
		45	**1**	0.7454	0.6

The boldface numbers indicate the best correlation coefficients

Table 3 Comparison results: robustness against additive noise attack

Model	α	σ^2	Corr. proposed	Corr. [21]	Corr. [6]
Camel	0.02	0.013	**1**	0.8783	0.8783
		0.015	**0.8783**	**0.8783**	0.8704
		0.017	**0.8704**	0.8704	0.8704
Elephant	0.02	0.0095	**1**	**1**	0.8783
		0.011	**0.8783**	0.8704	0.7333
		0.013	0.7333	**0.7454**	0.7333
Bunny	0.0002	0.013	**1**	**1**	0.8783
		0.015	0.8783	**1**	0.8704
		0.017	**0.8783**	0.8704	0.7746
Cow	0.01	0.0095	**1**	**1**	**1**
		0.012	**0.8783**	0.8704	1
		0.015	0.7333	**0.7746**	0.8704
Rocker-arm	0.0001	0.095	**1**	**1**	**1**
		0.012	0.8783	**1**	0.8783
		0.015	0.8704	**0.8783**	0.8783
Hand	0.09	0.0095	**1**	**1**	0.8783
		0.012	**0.8783**	**0.8783**	0.8704
		0.015	0.5919	**0.6**	0.5222

The boldface numbers indicate the best correlation coefficients

Table 4 Comparison results: robustness against compression attack

Model	α	No. of comp. basis	Corr. proposed	Corr. [21]	Corr. [6]
Camel	0.02	300	**1**	0.8783	0.8783
		250	**0.8783**	0.7746	0.6931
		200	**0.6831**	0.6	0.5222
Elephant	0.02	1700	**1**	0.6181	0.8783
		1500	**1**	0.5164	0.7746
		1300	**0.7746**	0.4229	0.6831
Bunny	0.0002	800	**1**	0.731	0.8783
		600	**1**	0.5164	0.8704
		500	**0.8783**	0.3133	0.5164
Cow	0.01	500	**1**	0.7746	0.5146
		400	**0.8783**	0.5164	0.333
		300	**0.7746**	0.4229	0.2582
Rocker-arm	0.0001	800	**1**	0.8783	0.8783
		600	**1**	0.6831	0.6831
		500	**0.8783**	0.3587	0.5164
Hand	0.09	1200	**1**	0.8783	0.8783
		1000	**0.8783**	0.7333	0.7333
		800	**0.7746**	0.4667	0.6181

The boldface numbers indicate the best correlation coefficients

Table 5 Comparison results: robustness against smoothing and simplification attacks

Model	α	Simpl. rate (%)	No. of iterations	Corr. proposed	Corr. [21]	Corr. [6]
Camel	0.02	20	9	**0.8783**	0.7746	0.6541
Elephant	0.02	30	8	**1**	0.8704	0.3456
Bunny	0.0002	35	7	**1**	0.6	0.522
Cow	0.01	20	9	**0.8783**	0.7746	0.5146
Rocker-arm	0.0001	35	8	**1**	0.8783	0.8704
Hand	0.09	30	25	**1**	**1**	0.8783

The boldface numbers indicate the best correlation coefficients

numbers of iterations were applied. Clearly our proposed scheme performs the best in terms of the robustness against the smoothing attack. An example of the smoothing attack is shown in Fig. 20. Resistance against the noise attack is shown in Table 3 where Gaussian random noise was added to each vertex of the watermarked model with three different standard deviations. Figure 21 shows the noise attack on different models. All the spectral techniques have good resistance against the noise attack because the watermark is embedded in all the spectral coefficients. To evaluate the robustness of the three techniques against the compression attack, we first simplified the mesh by reducing the number of faces to 10,000 for all the large the 3D models, then the 3D models are compressed using the algorithm proposed in [16]. The third column in Table 4 indicates the number of the basis functions used to compress the watermarked models. The correlation coefficients shown in Table 4 clearly demonstrate that our proposed scheme outperforms the existing techniques, and an example is shown in Fig. 22. To evaluate the robustness against smoothing and simplification attacks, we used various simplification rates and a fixed smoothing iteration number. Table 5 demonstrates that the proposed method performs better than the other techniques against the combination of simplification and smoothing attack. This better performance is in fact consistent with a variety of 3D models used for experimentation.

4.4 Computational complexity

The computational complexity and memory requirements of the watermark embedding process of our proposed scheme is the same as for the other two schemes proposed in [6,21]. The computation of the spectral basis functions of the Laplacian matrix is the most expensive and it is common to all the spectral domain schemes. The spectral analysis involves the computation of the basis functions of the Laplacian matrix of the 3D mesh. We partition the mesh into sub-meshes of n vertices each. The algorithm requires $\mathcal{O}(n^3)$ operations and stores $\mathcal{O}(n^2)$ elements of the calculated eigenvectors for straightforward implementations. Significant improvement could be further achieved by using the fast multi-resolution method [31] on the Laplacian matrix with an overall time complexity of $\mathcal{O}(n)$.

5 Conclusions

In this paper, we proposed a simple and computationally inexpensive watermarking methodology for embedding a watermark in the frequency domain of 3D models. The key idea is to encode a watermark vector repeatedly into the spectral coefficients of the compressed 3D mesh. A nonlinear visual error was used to test the perceptual quality of the watermarked 3D mesh. The performance of the proposed method was evaluated through extensive experiments that clearly showed a perfect resiliency against a wide range of attacks. For future work, we plan to analyze the relationship between the number of basis vectors used in the compression process, watermark length, mesh partition size, and strength factor to further improve the robustness against attacks.

References

1. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2001)
2. Benedens, O.: Geometry-based watermarking of 3-D polygonal models. IEEE Comput. Graph. Appl. **19**(1), 46–45 (1999)
3. Harte, T., Bors, A.: Watermarking 3d models. In: Proceedings of IEEE International Conference on Image Processing, pp. 661–664 (2002)
4. Ohbuchi, R., Masuda, H., Aono, M.: Watermarking three-dimensional polygonal models through geometric and topological modifications. IEEE J. Selected Areas Commun. **16**(4), 551–560 (1998)
5. Praun, E., Hoppe, H., Finkelstein, A.: Robust mesh watermarking. In: Proceedings of SIGGRAPH, pp. 49–56 (1999)
6. Ohbuchi, R., Takahashi, S., Miyasawa, T., Mukaiyama, A.: Watermarking 3-D polygonal meshes in the mesh spectral domain. In: Proceedings of Computer Graphics Interface, pp. 9–17 (2001)
7. Qiu, J.J., Ya, D.M., Jun, B.H., Sheng, P.Q.: Watermarking on 3D mesh based on spherical wavelet transform. J. Zhejiang Univ. Sci. **5**(3), 251–258 (2004)
8. Li, L., Pan, Z., Zhang, M., Ye, K.: Watermarking subdivision surfaces based on addition property of Fourier transform. In: Proceedings of International Conference on Computer Graphics and Interactive Technology, pp. 46–49 (2004)
9. Cotting, D., Weyrich, T., Pauly, M., Gross, M.: Robust watermarking of point-sampled geometry. In: Proceedings of International Conf.erence on Shape Modeling and Applications, pp. 233–242 (2004)
10. Kwon, K., Kwon, S., Lee, S., Kim, T., Lee, K.: Watermarking for 3D polygonal meshes using normal vector distributions of each patch. In: Proceedings of International Conference Image Process, pp. 499–502 (2003)
11. Zafeiriou, S., Tefas, A., Pitas, I.: Blind robust watermarking schemes for copyright protection of 3D mesh objects. IEEE Trans. Vis. Comput. Graph. **11**(5), 596–607 (2005)
12. Garcia, E., Dugelay, J.L.: Texture-based watermarking of 3-D video objects. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 853–866 (2003)
13. Cho, J.W., Prost, R., Jung, H.Y.: An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. IEEE Trans. Signal Process. **55**(1), 142–155 (2007)
14. Uccheddu, F., Corsini, M., Barni, M.: Wavelet-based blind watermarking of 3d models. In: Proceedings of ACM Multimedia and Security Workshop, pp. 143–154 (2004)
15. Wu, J., Kobbelt, L.: Efficient spectral watermarking of large meshes with orthogonal basis functions. Vis. Comput. **21**(8–10), 848–857 (2005)
16. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of SIGGRAPH, pp. 279–286 (2000)
17. Karypis, G., Kumar, V.: MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Version 4.0, University of Minnesota, Department of Computer Science (1998)
18. Ben Hamza, A., Krim, H.: Geodesic matching of triangulated surfaces. IEEE Trans. Image Process. **15**(8), 2249–2258 (2006)
19. Besl, J.B., Mckay, D.N.: A method for registration of 3D shapes. IEEE Trans. Pattern Anal. Machine Intell. **14**(2), 239–256 (1992)
20. Mian, A.S., Bennamoun, M., Owens, R.: A novel representation and feature matching algorithm for automatic pairwise registration of range images. Int. J. Comput. Vis. **66**(1), 19–40 (2006)
21. Ohbuchi, R., Mukaiyama, A., Takahashi, S.: A Frequency domain approach to watermarking 3D shapes. Comput. Graph. Forum **21**(3), 373–382 (2002)
22. Zhang, Y., Ben Hamza, A.: PDE-based smoothing for 3D mesh quality improvement. In: Proceedings of IEEE International Conference on Electro/Information Technology, pp. 334–337 (2006)
23. Rey, W.J.: Introduction to Robust and Quasi-robust Statistical Methods. Springer, Berlin (1983)
24. Alface, P.R., Craene, M.D., Macq, B.: Three-dimensional image quality measurement for the benchmarking of 3D watermarking schemes. In: Proceedings of Secu., Steganography, and Watermarking of Multimedia Contents, pp. 230–240 (2005)
25. Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J.J., Su, J.K.: Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. IEEE Commun. Mag. **39**(8), 118–126 (2001)
26. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Attacks on copyright marking systems. In: Proceedings of Workshop Info. Hiding, pp. 218–238 (1998)
27. Vollmer, J., Mencl, R., Muller, H.: Improved Laplacian smoothing of noisy surface meshes. In: Proceedings of EUROGRAPHICS, pp. 131–138 (1999)
28. Gumhold, S., Strasser, W.: Real time compression of triangle mesh connectivity. In: Proceedings of SIGGRAPH, pp. 133–140 (1998)
29. Isenburg, M., Snoeyink, J.: Mesh collapse compression. In: Proceedings of 12th Brazilian Symposium on Computer Graphics Image Process, pp. 27–28 (1999)
30. Garland, M., Heckbert, P.: Surface simplification using quadric error metrics. In: Proceedings of SIGGRAPH, pp. 209–216 (1997)
31. Guskov, I., Sweldens, W., Schroeder, P.: Multiresolution signal processing for meshes. In: Proceedings of SIGGRAPH, pp. 325–334
32. Cayre, F., Alface, P.R., Schmitt, F., Macq, B., Maître, H.: Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry. Signal Process. Image Commun. **18**(4), 309–319 (2003)