Spectral Graph-Theoretic Approach to 3D Mesh Watermarking

Emad E. Abdallah
Department of Computer Science
Concordia University, Montreal, Canada
ee_abdal@cs.concordia.ca

A. Ben Hamza and Prabir Bhattacharya
Concordia Institute for Information Systems Engineering
Concordia University, Montreal, Canada
hamza@ciise.concordia.ca, prabir@ciise.concordia.ca

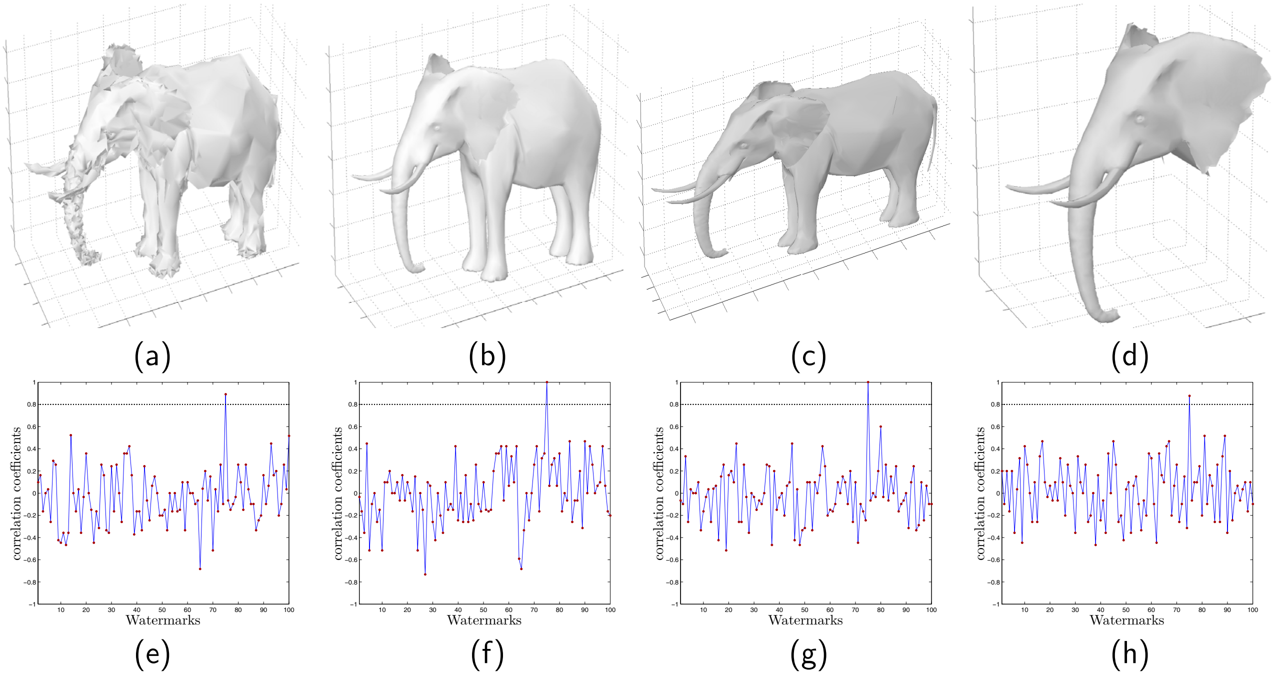


Figure 1: Watermarked elephant model with different attacks and their corresponding detector responses: for each each attack the correlation coefficient between the extracted watermark and 99 different random watermarks are shown (75 on X-axis is the correlation with the real watermark). (a), (e) Additive noise ($\sigma^2 = 0.007$), (b), (f) Low pass filter 9 iterations, (c), (g) scaling in X direction, (d), (h) cropping 1700 vertices and smoothing 7 iterations.

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.

Graphics Interface Conference 2007
28-30 May, Montréal, Canada

1 Introduction

Digital copyright protection of multimedia elements has long been the center of multimedia security research. Its importance is increasing rapidly in the field of computer graphics and multimedia communication because of the growing problem of the unauthorized duplication. Digital watermarking technology provides law enforcement officials with a forensic tool and an effective means of tracing and catching pirates. Watermarking refers to the process of adding a hidden structure, called a watermark, which carries either information about the owner of the cover or the recipient of the original data object. Whenever the rightful ownership is in question, the owner can extract the embedded watermark to make an assertion about the object. Such a watermark can be used for a wide range of applications including copyright protection, transaction tracing, copy protection, broadcast monitoring, data authentication, indexing, and medical safety [1–3].

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 [4–6] 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 [7–9] that are mainly based on multi-resolution mesh analysis (spectral decomposition and wavelet transform) and show good resistance against attacks. In [7] the generalized spread spectrum technique was proposed, and a set of scalar basis functions has been constructed over the mesh vertices using multi-resolution analysis where the watermark perturbs the vertices of each mesh along the direction of the surface normal, weighted by the basis functions. In [9] 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 [10] a watermarking scheme for subdivision surfaces has been presented. In [8] 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 [11] to watermark point-based 3D geometries. In [12] the normal vector distribution has been used. A blind watermarking scheme robust against affine transformation attacks was proposed in [13]. Watermarking of texture attributes has been proposed in [14]. Wavelet blind watermarking scheme has been proposed in [15] 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 proposed in [16]. In [17] the mesh Laplacian matrix was used to encode the 3D shape into a more compact representation by retaining the smallest eigenvalues and associated eigenvectors which 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.

The remainder of this paper is organized as follows. In section 2, we briefly review some background material and describe the spectral compression of the mesh geometry. In Section 3 we introduce the proposed approach and describe in detail the watermark embedding and extraction algorithms. In Section 4, we present some experimental results, and we show the robustness against the most common attacks. Finally, we conclude and point out future directions in Section 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 $\mathcal{V} = \{v_1, \ldots, v_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{t_1, \ldots, t_n\}$ is the set of triangles. Each edge $e_{ij} = [v_i, v_j]$ connects a pair of vertices $\{v_i, v_j\}$. Two distinct vertices $v_i, v_j \in \mathcal{V}$ are adjacent (written $v_i \sim v_j$) if they are connected by an edge $e_{ij} \in \mathcal{E}$. The neighborhood of a vertex v_i is the set $v_i^\star = \{v_j \in \mathcal{V} : v_i \sim v_j\}$. The degree d_i of a vertex v_i is the cardinality of $v_i^\star$. Let $v_i = (x_i, y_i, z_i) \in \mathcal{V}$, $1 \leq i \leq m$, then the mesh vertex matrix V is the $m \times 3$ matrix whose i^{th} row is the vector 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: $L = A - D$, where A is the adjacency matrix between the vertices, defined by:

$$A_{ij} = \begin{cases} 1 & \text{if } v_i \sim v_j \\ 0 & \text{otherwise} \end{cases}$$

and D is the $m \times m$ diagonal matrix whose (i,i) entry is d_i. Figure 2 illustrates an example of a 3D triangle meshes and its sparse Laplacian matrices.

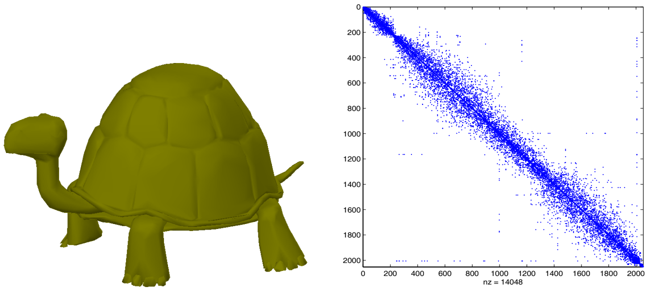


Figure 2: 3D triangle mesh and its Laplacian matrix.

2.3 Spectral Mesh Compression

In [17] 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 L is given by

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

where $B = (b_1\, b_2\, \ldots b_m)$ is an orthogonal matrix whose columns b_i are the eigenvectors of L which we refer to as Laplacian basis functions, and $\Lambda = \mathrm{diag}\{\lambda_i : 1 \le i \le m\}$ is a diagonal matrix of the eigenvalues of L arranged in increasing order of magnitude. We express the mesh vertex matrix in the subspace spanned by the Laplacian matrix eigenvectors as follows:

$$V^T = C^T B^T = \sum_{i=1}^{m} c_i^T b_i^T \tag{2}$$

where $C = (c_1\, c_2\, \ldots c_m)^T$ is an $m \times 3$ matrix of the spectral coefficient vectors, that is, $C = B^T V$ where C is the projection of the mesh vertex matrix onto the Laplacian basis vectors. Moreover, Eq. (2) can be written as:

$$V^T = \underbrace{\sum_{i=1}^{r} c_i^T b_i^T}_{\text{compressed}} + \sum_{i=r+1}^{m} c_i^T b_i^T = C_r^T B_r^T + \sum_{i=r+1}^{m} c_i^T b_i^T \tag{3}$$

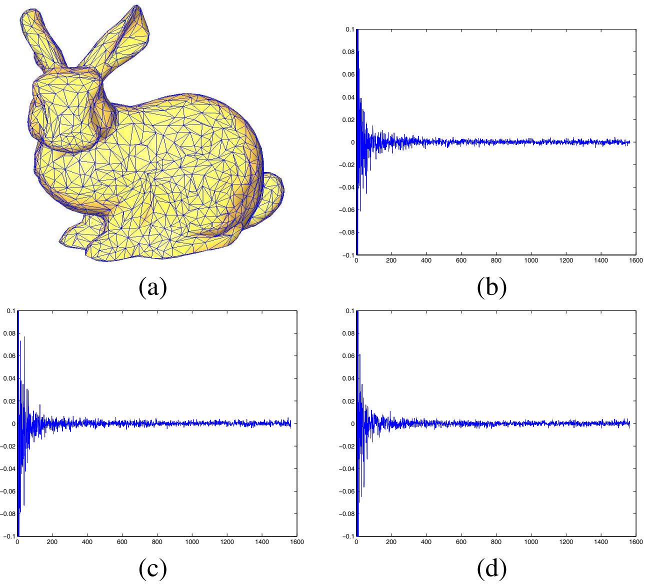
where r is usually chosen to be smaller than m, and hence this yields a compressed mesh version $\mathbb{M}_r$ of the original mesh $\mathbb{M}$ with a very small loss in the mesh quality. The matrix $B_r = (b_1\, b_2\, \ldots b_r)$ contains the spectral basis vectors, and the matrix $C_r = (c_1\, c_2\, \ldots c_r)^T$ contains the spectral coefficient vectors. If we rewrite V and C in the form of 3-column matrices, that is

$$V = (v_x\; v_y\; v_z) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix}$$

and

$$C = (c_x \quad c_y \quad c_z) = \begin{pmatrix} c_{x_1} & c_{y_1} & c_{z_1} \\ c_{x_2} & c_{y_2} & c_{z_2} \\ \vdots & \vdots & \vdots \\ c_{x_m} & c_{y_m} & c_{z_m} \end{pmatrix},$$

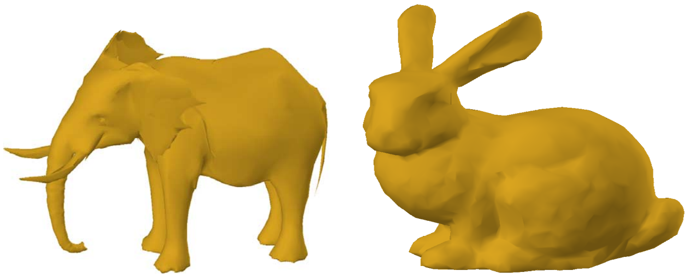
The spectral coefficients in the x, y, and z-dimension are given by $c_x = B^T v_x$, $c_y = B^T v_y$, and $c_z = B^T v_z$ respectively, see Figure 3 for an example. Figure 4 shows two examples of the mesh compression results using Laplacian-based method with 500 basis functions.



(a) (b)

(c) (d)

Figure 3: (a) 3D rabbit 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

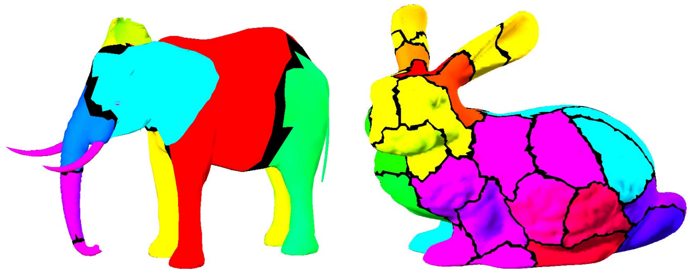
Figure 4: Spectral compression of the 3D models using Laplacian compression (a) Elephant model (b) Rabbit model.

2.4 Mesh Partitioning

The computation of the eigenvalues and the eigenvectors of a large $m \times m$ Laplacian matrix is prohibitively expensive $O(m^3)$. 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 approach we used the MeTiS software [18] for mesh partitioning, and we used sub-meshes of 500 vertices on average as illustrated in Figure 5.

3 PROPOSED METHOD

In this section, we describe the main steps of the proposed watermark embedding and extraction algorithms, Figure 6 and Figure 8 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



(a) (b)

Figure 5: MeTis mesh partitioning. Each sub-mesh is colored by a random color. Black triangles represent edge cuts. (a) Elephant model: 4067 vertices, 8 sub-meshes. (b) Rabbit model: 20100 vertices, 40 sub-meshes.

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.

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 $k << n$. For all sub-meshes the watermark embedding process consists of the following steps:
1. Compute the Laplacian matrix L of size $n \times n$.
2. Compute the eigenvalues and the associated eigenvectors (basis functions) of L.
3. Project the mesh vertices onto the basis functions to get the spectral coefficients matrix $C = B^T V$ of the original sub-mesh $\mathbb{S}$.
4. Use 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 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 C_r by the watermark sequence W_d. So,

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

where $\widehat{C}_r$ 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. Thus,

$$V_{W_r}^T = \widehat{C}_r^T B^T = \sum_{i=1}^{r} \widehat{c}_{i_r}^T b_i^T$$

where $V_{W_r}^T$ 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

$$V_W^T = V_{W_r}^T + \sum_{i=r+1}^{n} \bar{c}_i^T b_i^T \tag{5}$$

where $\overline{C} = \{\bar{c}_i\}_{r+1}^n$ is the spectral coefficients matrix of the high frequency basis functions. Figure 7 shows two different 3D models with their corresponding watermarked meshes.

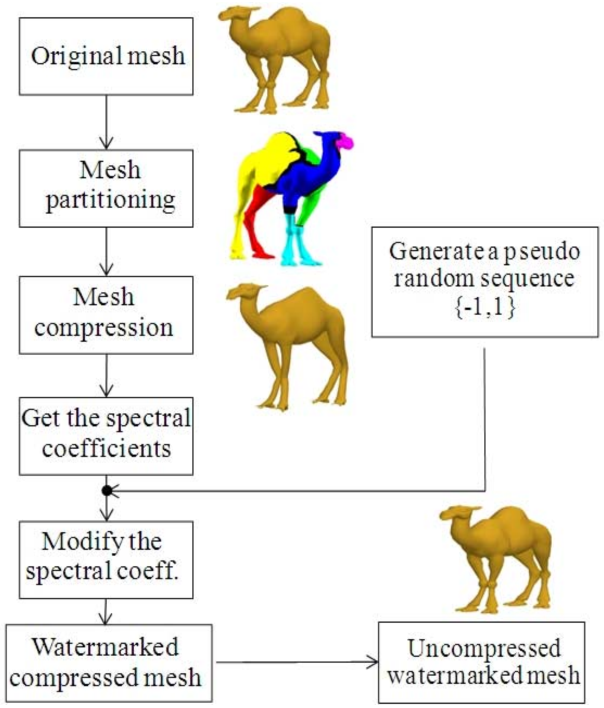

Original mesh
Mesh partitioning
Mesh compression
Get the spectral coefficients
Generate a pseudo random sequence {-1,1}
Modify the spectral coeff.
Watermarked compressed mesh
Uncompressed watermarked mesh

Figure 6: Watermark embedding process.

3.2 Watermark Extraction Process

We provide a private watermarking scheme, that is, the original unwatermarked object is necessary for the extraction process. Let the original unwatermarked mesh be $\mathbb{M}$ and the watermarked probably attacked mesh be $\widehat{\mathbb{M}}$.

i) Mesh Registration

Before applying the extraction process, 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.

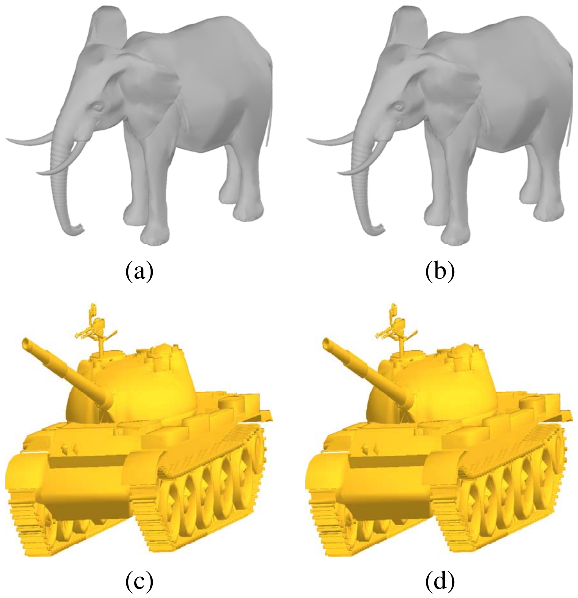

(a) (b)
(c) (d)

Figure 7: Original 3D models (a)-(c) and their corresponding watermarked models (b)-(d). Elephant model with (4076 vertices, 7999 faces) and tank model with (15186 vertices, 13902 faces).

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 versions of each.
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 vectors 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.

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

Model	# vertices	# faces	# patches	# watermarks
Camel	4001	8050	7	750
Rabbit	20100	39999	40	3768
Max Planck	5040	10067	10	945
Elephant	4067	7999	8	762
Tank	15186	13902	31	2847
Mesh part	2496	5000	4	468

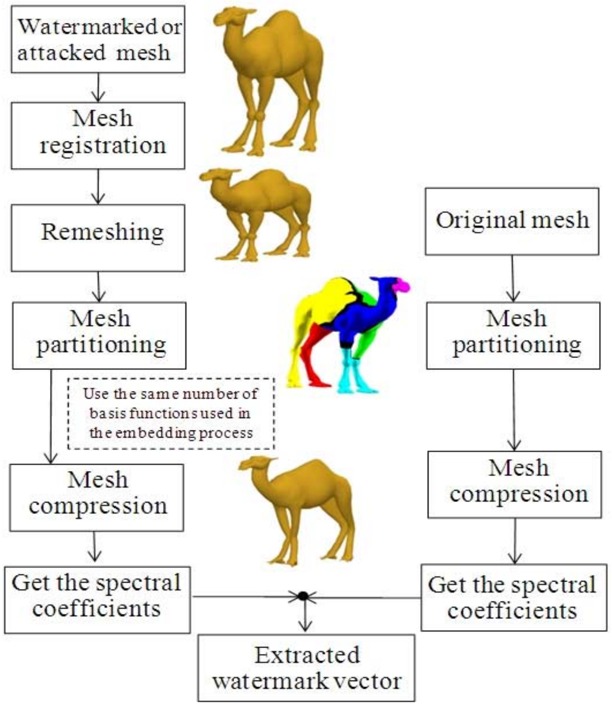

Watermarked or attacked mesh
Mesh registration
Remeshing
Mesh partitioning
Use the same number of basis functions used in the embedding process
Mesh compression
Get the spectral coefficients
Original mesh
Mesh partitioning
Mesh compression
Get the spectral coefficients
Extracted watermark vector

Figure 8: Watermark extraction process.

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. We conducted experiments to test the robustness against attacks.

4.1 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 [22, 23]. 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. A watermark sequence of 16 bits is generated randomly of $\{-1,1\}$. In the experiments we display the attacked models with the detector response for the real watermark, and 99 randomly other generated watermarks. For all the detector response figures the correlation between the original watermark and the extracted watermark is located at 75 on the X-axis and the dotted line at 0.8 on the Y-axis represents the threshold. The threshold is chosen manually to decrease false-positive (presenting incorrectly the watermark in the model) and false-negative alarm (falling to detect the watermarked model). If the correlation is larger than 0.8, then the watermark is present. In all the experiments the strength factors that have been used are 0.01 for the Max Planck model, 0.009 for the camel model, and 0.02 for the elephant model.

4.1.1 Additive Random Noise

In order to test the robustness of the watermark, an additive Gaussian noise was added to the watermarked mesh by summing a random vector to each vertex in the model. See Figure 9(a) 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 Figure 9(c). The watermark is lost when we increased the noise $(\sigma^2 = 0.0045)$ for the Max Planck model.

4.1.2 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 [24] 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 9(b) depicts the attacked Max Planck model by 7 smoothing iterations, and Figure 9(d) shows the detector response. As can be seen, the mesh is significantly smoothed but the watermark is still perfectly detectable.

4.1.3 Geometric Transformations

These are the simplest attacks used to test the watermark detectors. The proposed algorithm is robust against geometric attacks because the transformations applied to the mesh can be inverted using mesh registration. In Figure 10(a) 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^o. Figure 10(c) depicts the watermark extraction response after the registration process was applied. Clearly the detector is still able to recover the watermark.

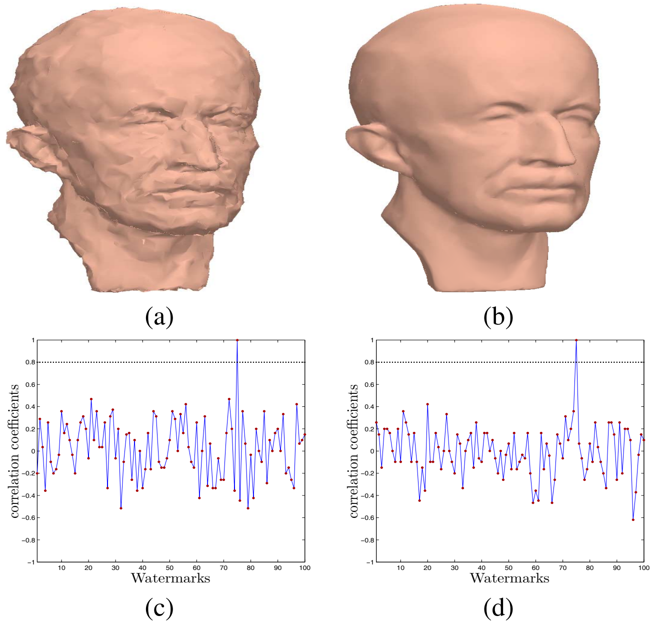


Figure 9: Robustness against random noise and Laplacian smoothing attacks. (a) Max Planck model with $(\sigma^2 = 0.0035)$ additive noise. (b) Max Planck model after (7 iterations) of the low pass filter. (c), (d) Detector responses for (a), (b) respectively.

4.1.4 Mesh Compression

Mesh compression has recently become one of the most effective attacks because the new compression techniques [17, 25] 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 [17]. The proposed method is robust against compression because the watermark is embedded in the spectral coefficient of the compressed mesh. Figure 10(b) depicts the compressed Max Planck model constructed with 3000 basis functions from the original mesh of 5040 basis functions. The detector response is shown in Figure 10(d).

4.1.5 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

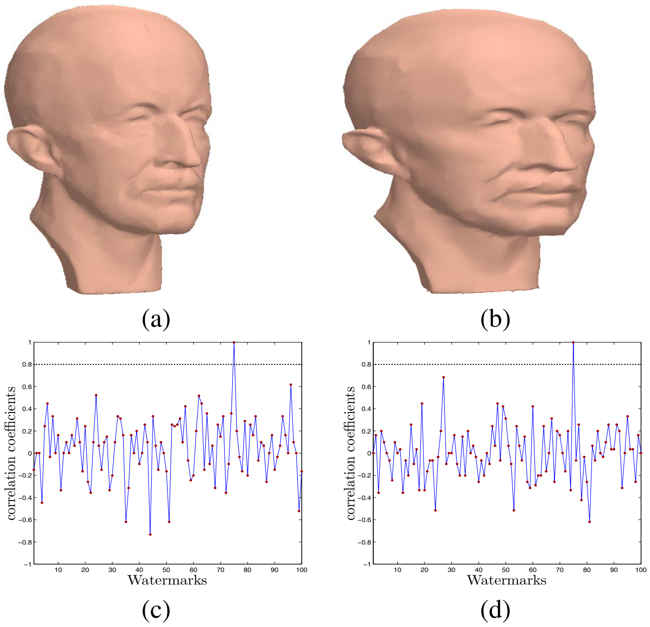


Figure 10: Robustness against geometric transformation and compression attacks. (a) Max Planck model is scaled in Z direction by factor of 2 then rotated by 20^o around Y-axis. (b) Compressed Max Planck model of 3000 basis functions. (c), (d) Detector responses for (a), (b) respectively.

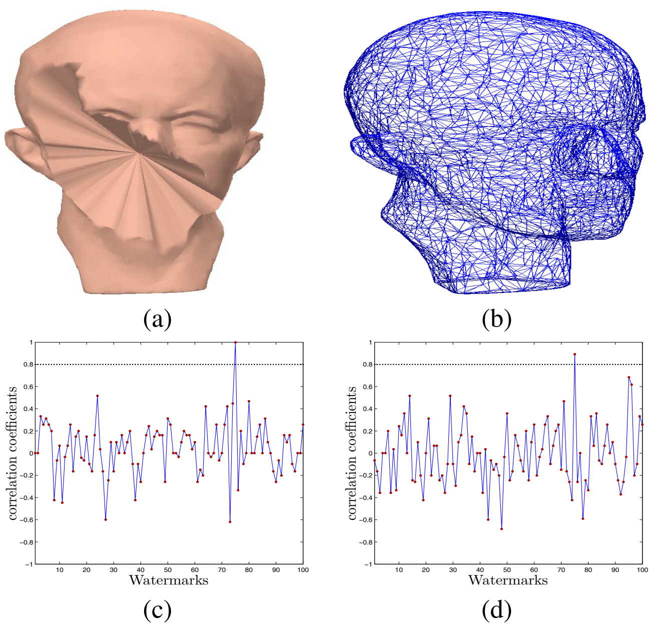


Figure 11: Robustness against cropping and mesh simplification attacks. (a) Cropped 600 vertices from Max Planck model (b) Max Planck simplified down to 2502 vertices and 5000 faces. (c), (d) Detector responses for (a), (b) respectively.

from the deteriorated cropped mesh. Figure 11(a) 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 Figure 11(c).

4.1.6 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 Figure 11(b) for the simplified Max Planck model. The mesh is simplified down from 5040 vertices and 10067 faces to 2502 vertices and 5000 faces. Our proposed method is robust against the simplification attack because of the remeshing process. The detector response for the attacked mesh in Figure 11(b) is illustrated in Figure 11(d).

We also tested the performance of our proposed algorithm using a combination of the previous attacks. Figure 12 (a,b) show the watermarked Max Planck model with multiple attacks. In Figure 12(a) 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 mesh. Figure 12(b) 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 Figure 1.

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. 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] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, San Francisco, 2001.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp. 1079-1107, 1999.

[3] N. Memon and P. Wong, "Digital watermarks: protecting multimedia content," *Communications ACM*, vol. 47, no. 7, pp. 35-43, 1998.

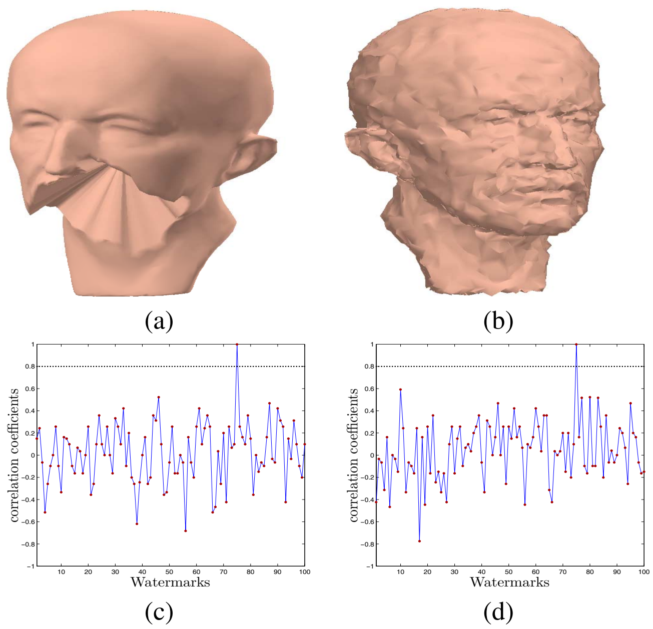
[4] O. Benedens, "Geometry-based watermarking of 3-D polygonal models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46-45, 1999.

[5] T. Harte and A. Bors, "Watermarking 3d models," *Proc. IEEE Internat. Conf. Image Process.*, pp. 661-664, 2002.

[6] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Selected Areas in Comm.*, vol. 16, no. 4, pp. 551-560, 1998.

[7] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," *Proc. SIGGRAPH*, pp. 49-56, 1999.

[8] R. Ohbuchi, S. Takahashi, T. Miyasawa, and A. Mukaiyama, "Watermarking 3-D polygonal meshes in the mesh spectral domain," *Proc. Computer Graphics Interface*, pp. 9-17, 2001.



(a) (b)

(c) (d)

Figure 12: 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), (b) respectively.

[9] J. J. Qiu, D. M. Ya, B. H. Jun, and P. Q. Sheng, "Watermarking on 3D mesh based on spherical wavelet transform," *J. Zhejiang Univ. Sci.*, vol. 5, no. 3, pp. 251-258, 2004.

[10] L. Li, Z. Pan, M. Zhang, and K. Ye, "Watermarking subdivision surfaces based on addition property of Fourier transform," *Proc. Internat. Conf. Computer Graphics and Interactive Tech.*, pp. 46-49, 2004.

[11] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," *Proc. Internat. Conf. Shape Modeling and Applicat.*, pp. 233-242, 2004.

[12] K. Kwon, S. Kwon, S. Lee, T. Kim, and K. Lee, "Watermarking for 3D polygonal meshes using normal vector distributions of each patch," *Proc. Internat. Conf. Image Process.*, pp. 499-502, 2003.

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

[14] E. Garcia and J. L. Dugelay, "Texture-based watermarking of 3-D video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 853-866, 2003.

[15] F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3d models," *Proc. ACM Multimedia and Security Workshop*, pp. 143-154, 2004.

[16] J. Wu and L. Kobbelt, "Efficient spectral watermarking of large meshes with orthogonal basis functions," *The Visual Computer*, vol. 21, no. 8-10 pp. 848- 857, 2005.

[17] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," *Proc. SIGGRAPH*, pp. 279-286, 2000.

[18] G. Karypis and V. Kumar, "MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fillreducing orderings of sparse matrices," Version 4.0, Univ. Minnesota, Dept. Computer Sci., 1998.

[19] J. B. Besl and D. N. Mckay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239- 256, 1992.

[20] A. S. Mian, M. Bennamoun, and R. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images", *Internat. J. of Computer Vision*, vol. 66, no. 1, pp. 19-40, 2006.

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

[22] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, " Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE Comm. Mag.*, vol. 39, no. 8, pp. 118-126, 2001.

[23] F. A. P. Petitcolas, R. J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," *Proc. Workshop Info. Hiding*, pp. 218-238, 1998.

[24] J. Vollmer, R. Mencl, and H. Muller, "Improved Laplacian smoothing of noisy surface meshes," *Proc. EUROGRAPHICS*, pp. 131-138, 1999.

[25] S. Gumhold and W. Strasser, "Real time compression of triangle mesh connectivity," *Proc. SIGGRAPH*, pp. 133-140, 1998.