Local patch blind spectral watermarking method for 3D graphics

Ming Luo[1], Kai Wang[2], Adrian G. Bors[1], and Guillaume Lavoué[2]

[1] Department of Computer Science, University of York, York YO10 5DD, UK
[2] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France

Abstract. In this paper, we propose a blind watermarking algorithm for 3D meshes. The proposed algorithm embeds spectral domain constraints in segmented patches. After aligning the 3D object using volumetric moments, the patches are extracted using a robust segmentation method which ensures that they have equal areas. One bit is then embedded in each patch by enforcing specific constraints in the distribution of its spectral coefficients by using Principal Component Analysis (PCA). A series of experiments and comparisons with state-of-the-art in 3D graphics watermarking have been performed; they show that the proposed scheme provides a very good robustness against both geometry and connectivity attacks, while introducing a low level of dis torsion.

1 Introduction

During the last decade, various 3-D watermarking methods have been developed. Watermarking algorithms can be classified as being in the spatial domain [1,2,3] or the frequency domain [4,5]; they also can be classified as relying on local constraints [1] or as being statistical in nature [2,3]. A survey of existing 3D watermarking algorithms was carried out in [6]. Most of the robust blind methods developed so far are in the spatial domain. Methods in the frequency domain include those using wavelets as well as those using spectral decomposition [4,5]. The spectral transform consists of the eigen-decomposition of the Laplacian matrix for a given mesh. The resulting spectral coefficients consist of a unique representation. The spectral decomposition is reversible and the graphical object mesh can be entirely reconstructed from the spectral coefficients. Spectral methods were firstly introduced in graphics by Karni and Gotsman [7] for the purpose of mesh compression. Ohbuchi [4] proposed a non-blind method to embed watermarks in the spectral domain based on Karni's analysis. Lavoué et al. [8] and Cotting et al. [9] improved Ohbuchi's algorithm for subdivision mesh and point sampling distributions, respectively. All these algorithms use the low-frequency coefficients as the message carrier and are non-blind. Blind and robust spectral domain watermarking have been proposed in [5,10,11]. The algorithm described in [10] has a very low bit-capacity while a set of specific constraints are enforced onto the distributions of spectral coefficients for information embedding in [11].

The advantage of the spectral domain mainly relies in the fact that the watermark information is spread over the object in such a way that it is very hard

to determine its presence, thus increasing its security. Usually, existing spectral domain algorithms are not that robust as the spatial domain algorithms. Another disadvantage for spectral domain watermarking is the computational complexity required for the eigendecomposition of the Laplacian matrix corresponding to meshes containing a large number of vertices, which is actually the case with most graphical objects [11].

In this paper we propose a novel robust blind spectral watermarking method which consists of applying spectral decomposition locally, in well defined patches of the graphical object. The proposed methodology has the following stages. Firstly, the object is robustly aligned along the principal axis calculated using the analytic volumetric moments. Then the object is decomposed into patches (*i.e.* connected spatial regions) of equal areas defined along the first and second principal axes. Lastly, the spectral analysis is performed on each patch and the spectral coefficients are extracted. The watermark insertion is done by embedding specific constraints in the distribution of the spectral coefficients. The constraints are enforced by an embedding function which relies on the principal component analysis (PCA) of the spectral coefficients. The distribution of the spectral coefficients is constrained to a sphere when embedding a bit of zero and to a squashed ellipsoid when embedding a bit of one. The reminder of this paper is organized as follows. In Section 2 we describe the volumetric method for aligning the 3D graphical object, while in Section 3 we describe the algorithm for generating the equal area patches. In Section 4 the spectral graph theory is briefly introduced, while Section 5 gives the details of the proposed watermark insertion and extraction based on spectral coefficients analysis. The experimental results and comparison with the state of the art are provided in Section 6, while the conclusions of this study are drawn in Section 7.

2 Robust object alignment

We propose to use a robust alignment scheme called volume moment alignment which was proposed in [12]. The volume moments of a 3D object are defined as:

$$M_{pqr} = \int\int\int x^p y^q z^r \rho(x,y,z) d_x d_y d_z \tag{1}$$

where p, q, r are moment orders, and $\rho(x,y,z)$ is the volume indicator function (it equals to 1 if (x,y,z) is inside the mesh and to 0 otherwise). For a triangular face $f_i = \{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}\} = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}), (x_{i3}, y_{i3}, z_{i3})\}$ on a mesh object, the moments are defined as :

$$\begin{aligned}
M_{000}^{f_i} &= \tfrac{1}{6}|x_{i1}y_{i2}z_{i3} - x_{i1}y_{i3}z_{i2} - y_{i1}x_{i2}z_{i3} + y_{i1}x_{i3}z_{i2} + z_{i1}x_{i2}y_{i3} - z_{i1}x_{i3}y_{i2}| \\
M_{100}^{f_i} &= \tfrac{1}{4}(x_{i1} + x_{i2} + x_{i3}) \cdot M_{000}^{f_i} \\
M_{200}^{f_i} &= \tfrac{1}{10}(x_{i1}^2 + x_{i2}^2 + x_{i3}^2 + x_{i1}x_{i2} + x_{i1}x_{i3} + x_{i2}x_{i3}) \cdot M_{000}^{f_i} \\
M_{110}^{f_i} &= \tfrac{1}{10}(x_{i1}y_{i1} + x_{i2}y_{i2} + x_{i3}y_{i3} + \tfrac{x_{i1}y_{i2}+x_{i1}y_{i3}+x_{i2}y_{i1}+x_{i2}y_{i3}+x_{i3}y_{i1}+x_{i3}y_{i2}}{2}) \cdot M_{000}^{f_i}
\end{aligned} \tag{2}$$

In fact it corresponds to the moment of the tetrahedron linking this face to the coordinate system origin. The global moments of a mesh are obtained by summing

these elementary moments over all the facets (with the appropriate contribution sign). The complete set of explicit volume moment functions can be found in [13]. The object centre is defined as $\mu = (M_{100}/M_{000}, M_{010}/M_{000}, M_{001}/M_{000})$, and the 3×3 matrix of the second order moments of the 3D object is constructed as:

$$\mathbf{\Psi} = \begin{pmatrix} M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{pmatrix} \tag{3}$$

The principal axes of the object are the eigenvectors obtained by applying eigendecomposition to the covariance matrix $\mathbf{\Psi}$:

$$\mathbf{\Psi} = \mathbf{W}^T \mathbf{\Delta} \mathbf{W} \tag{4}$$

where $\mathbf{\Delta} = \{\delta_1, \delta_2, \delta_3\}$ is the diagonal matrix containing the eigenvalues assuming $\delta_1 > \delta_2 > \delta_3$ and $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3]^T$ is the matrix whose columns are the eigenvectors of $\mathbf{\Psi}$. The eigenvalues $\{\delta_1, \delta_2, \delta_3\}$ characterize the extension of the object along its principal axes whose directions are defined by the corresponding eigenvectors. In order to define a unique alignment, we propose two constraints. Firstly, the three axes must conform the right hand rule such that the direction of the third axis will be well defined as the cross product of the first two. Furthermore, the valid alignment satisfies the condition that the third order moments M_{300} and M_{030} of the rotated object are positive. By following these constraints the principal axis alignment is unique [12] and much more robust than the alignment produced by the moments using the vertex coordinates.

3 Object patch generation

Watermarking in spectral domain owns a series of advantages including increased watermark key security and good watermark imperceptibility. However, as shown in [11], the application of blind spectral watermarking is limited to rather small graphical objects due to the high computational complexity requirements of spectral decomposition for large meshes. In this study in order to apply the spectral algorithm to a large 3D object, we propose to split it into segments (*i.e.* spatial regions) so that each segment is used for carrying one bit of message. Note that one segment can be used to embed more than one bit, but in this paper, we only embed one bit for the sake of aiming for the highest robustness. After aligning the graphical object as described in Section 2 we trim away, for the further watermark embedding usage, the ends of the object as defined along its principal axis $\mathbf{w}_1$. In this way we increase the watermark security. Then the trimmed object is split into layers which are defined by planes perpendicular to $\mathbf{w}_2$, the second principal axis. Finally, the vertices and triangles in each layer are divided into connected patches of equal area in a direction along the first principal axis $\mathbf{w}_1$. All these steps are detailed below.

Let us consider x_{max} and x_{min} the maximum and minimum value along the first principal axis $\mathbf{w}_1$. $\alpha \in [0, 1/2]$ is a value generated by the secret key which is used for trimming the extremities of the object. We define two boundary values

x_1 and x_2 on the line, passing through the object center and in the direction of the first principal axis $\mathbf{w}_1$, such that:

$$\begin{cases} B_{min} = (x_{max} - x_{min}) \cdot \alpha + x_{min} \\ B_{max} = (x_{max} - x_{min}) \cdot (1 - \alpha) + x_{min} \end{cases} \tag{5}$$

All the vertices whose x coordinates are outside this range, *i.e.*, $v_x < B_{min}$ and $v_x > B_{max}$ will be excluded from the watermark embedding process. Let us define the trimmed object $\mathcal{O}_T$ as:

$$\mathcal{O}_T = \{\mathbf{v} | B_{min} \leq v_x < B_{max}, \ \ \forall \mathbf{v} \in \mathcal{O}\} \tag{6}$$

The trimmed object area considered for watermarking, denoted as A_t, is defined as the sum of all polygons, usually triangles, which are located on the surface of the trimmed object $\mathcal{O}_T$.

$\mathcal{O}_T$ is then split into κ layers defined by planes perpendicular on the second principal axis, *i.e.* $\mathbf{w}_2$. κ is chosen as a function depending on the number of bits M to be embedded and on the geometrical characteristics of the graphical object such as its aspect ratio given by δ_1/δ_2 obtained from (4). Thus, there are $\kappa + 1$ boundary values defined as:

$$y_i = y_{min} + \frac{y_{max} - y_{min}}{\kappa} \cdot i \tag{7}$$

where $i = 0, \ldots, \kappa$. The vertices of the trimmed object are thus split into κ layers and for each layer we have:

$$L_i = \{\mathbf{v} | y_{i-1} \leq \mathbf{v}_y < y_i, \ \ \forall \mathbf{v} \in \mathcal{O}_T\} \tag{8}$$

where $i = 1, \ldots, \kappa$. These layers are then divided such that to obtain a set of N patches of equal areas.

The desired area for each patch is calculated according to the following:

$$A_p = \frac{A_t}{N} \tag{9}$$

where $N > M$ and M is the number of bits to be embedded in the object. The reason for which more patches are generated than the number of bits to embed is because a segment may be eliminated if its area is smaller than a pre-defined threshold while we also aim to increase the watermark security by deliberately excluding certain patches. For example, the last patch of each layer is likely to have an area smaller than A_p and thus if watermarked it may result into a skewed distributed embedding capacity onto the 3D object surface.

Next, we sort all vertices of all layers L_i in ascending order of their x coordinates, *i.e.* along the first principal axis $\mathbf{w}_1$. Vertices are then iteratively added into a patch from left to right of the sorted sequence; when the area A_p is achieved for the current growing patch, a new patch is initiated. Let us denote $\mathcal{P}_j$ as the j^{th} patch generated, where $j = 1, \ldots, N$. The first M patches $\mathcal{P}_1, \ldots, \mathcal{P}_M$ are used for watermarking. The patches to be watermarked can be picked up randomly according to the watermark key. This patch segmentation mechanism is summarized in the Algorithm 1.

Algorithm 1 Patch Segmentation Algorithm

```
seg = 0; // Patch index
for i = 1 to κ do
    Let V be the sorted sequence of ∀v ∈ L_i in ascending order of the x coordinate
    A = 0;
    for j = 1 to |L_i| do
        v_j = V[j];
        Add v_j into patch P_seg;
        for all Neighbouring face f incident to v_j do
            if f is not processed and each vertex of f has been assigned to a patch then
                Calculate the area A_in inside the layer L_i;
                Increment the area accumulator A+ = A_in;
                if A > A_P then
                    seg++; // Move to the next patch;
                    Move v_j to patch P_seg
                    A = A − A_p // Residue area is assigned to the next patch
                end if
            end if
        end for
    end for
    Mark all v in the last patch of the layer L_i as −1.
end for
```

When one triangle is crossing several boundaries of layer planes, that triangle will be split and only the area of its section which is within the layer L_i will be accounted into the current patch area. For example, as shown in Fig. 1, triangle $\triangle ABC$ is located at the intersection of two different layers; B_{max} indicates the trimming boundary value. Only the area of the red region, *i.e.* polygon $ADEF$ is accumulated in the current patch area. Ideally, one segment should contain only one compact 3D patch surface. However, if an object has a complex topology, *e.g.* the graphical object contains many holes, one segment may include several small and isolated patches. Watermarking such discontinuous patches will result into visible distortions and may cause visible artifacts on the graphical object surface following spectral watermarking. Therefore, those patches which contain discontinuous areas that are smaller than a predefined threshold, resulting after segmentation, are removed from the further watermark embedding process.

There are several advantages for the proposed layer segmentation algorithm. Firstly, the patches generated using this algorithm are highly secure. As it can be observed the percentage of the trimmed extremities is generated according to a secret key and it is therefore impossible to recover the patches without the knowledge of this secret key. By increasing the number of layers we are more likely to achieve a superior patch compactness. Patches which are closer to a square-like shape are more appropriate for spectral watermarking since they provide higher area compactness and an increased connectivity which are both desired by a watermark embedding procedure. On the contrary, if only one layer

is used, the algorithm splits the object into narrow strips which contain a lower level of mesh connectivity. By adjusting the parameter κ, we therefore have the flexibility to adjust the size and the shape of the patches.

According to the experimental results the proposed patch segmentation algorithm is robust against most of the mesh attacks including additive noise, mesh simplification and Laplacian smoothing. Moreover, the process of watermarking the 3D object will have no influence on the graphical object segmentation procedure in the detection stage. Such a robustness is a very strong asset for a graphical object watermarking application. The proposed algorithm produces patches with equal areas and each patch will carry only one watermark bit, resulting into a high robustness. Two examples of patch segmentation for the Venus head graphical object are illustrated in Figs 2(a) and 2(b), for one layer and three layers, respectively. From these figures it is clear that three layers segmentation produces more compact patches than a single layer segmentation.

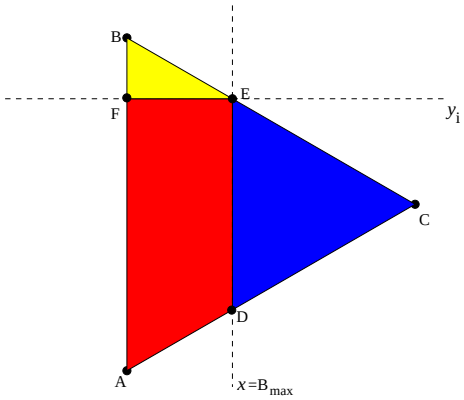


Fig. 1. An example of a mesh triangle splitting over different patches.

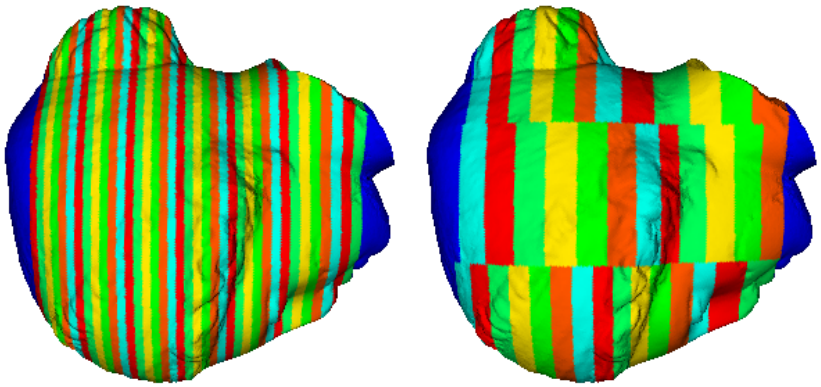
4 Spectral decomposition of mesh patches

A patch $\mathcal{P}_i$ consists of a set of vertices $\{\mathbf{v}_j, j = 1, \ldots, |\mathcal{P}_i|\}$ where $|\mathcal{P}_i|$ is the number of the vertices within the patch $\mathcal{P}_i$, and a set of edges characterizing the connectivity information. The Laplacian matrix $\mathbf{L}^i$ is calculated as the difference between the degree matrix and the adjacency matrix and has the following entries:

$$L^i_{j,k} = \begin{cases} |\mathcal{N}(\mathbf{v}_j)| & \text{if } j = k \\ -1 & \text{if } j \neq k \text{ and } \mathbf{v}_j \text{ adjacent to } \mathbf{v}_k \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $|\mathcal{N}(\mathbf{v}_j)|$ represents the degree of the vertex $\mathbf{v}_j$ (the number of neighbouring vertices $\mathbf{v}_k \in \mathcal{N}(\mathbf{v}_j)$). Then the Laplacian matrix is eigen-decomposed as :

$$\mathbf{L}^i = \mathbf{q}_i^T \Omega_i \mathbf{q}_i \tag{11}$$



(a) One layer $\kappa = 1$ (b) Three layers $\kappa = 3$

Fig. 2. Patch segmentation of the Venus head graphical object. The blue regions represent the trimmed extremities along the first principal axis $\mathbf{w}_1$.

where Ω_i is a diagonal matrix containing the eigenvalues of the Laplacian, and $\mathbf{q}_i$ is a matrix containing its eigenvectors. The eigenvectors of $\mathbf{L}^i$ represent an orthogonal basis and the associated eigenvalues are considered as frequencies. In the following we assume that $\mathbf{q}_i$ are sorted according to the ascending order of their corresponding eigenvalues from the diagonal matrix $\mathbf{\Omega}_i$. The spectrum is provided by the projections of each vertex coordinate on the directions defined by the basis function $\mathbf{q}_i$. The spectral coefficients $\mathbf{C}_i$ are calculated as:

$$\mathbf{C}_i = \mathbf{q}_i \mathbf{V}_i \tag{12}$$

where $\mathbf{V}_i$ is the set of spatial coordinates of the vertices of the patch.

The transformation is reversible and the patch vertices can be recovered as:

$$\mathbf{V}_i = \mathbf{q}_i^T \mathbf{C}_i \tag{13}$$

5 Watermarking using PCA of the spectral coefficients

5.1 Watermark Embedding

The spectral coefficients can be divided into "low frequency" and "high frequency". The low frequency reflects the large scale information of the 3D object while the high frequency corresponds to the details of the object. Changing the low frequency coefficients may result in severe deformation and the shearing of the object. In contrast, changing the high frequency could introduce noisy effects on the object surface. In this paper, we propose to embed the watermark in the high frequency coefficients so as to minimize the resulting object distortion. In the following we consider the distribution of the highest 70% of the spectral coefficients for watermarking. The high frequency coefficients for each

patch form a point cloud in the 3D space. The shape of this point cloud distribution is described by using Principal Component Analysis (PCA). The mean and covariance matrix of each set of points are calculated as:

$$\mu_i = \frac{\sum_{j=1}^{n} \mathbf{C}_{i,j}}{n} \tag{14}$$

$$\Sigma_i = \frac{1}{n} \sum_{j=1}^{n} (\mathbf{C}_{i,j} - \mu_i)^T (\mathbf{C}_{i,j} - \mu_i) \tag{15}$$

where n is the number of frequency coefficients. The covariance matrix Σ_i is decomposed as:

$$\Sigma_i = \mathbf{U}_i^T \Lambda_i \mathbf{U}_i \tag{16}$$

where Λ is the diagonal matrix containing the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ where we assume $\lambda_1 > \lambda_2 > \lambda_3$. $\mathbf{U}_i$ is the transformation matrix whose columns are the eigenvectors of Σ_i. The eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ determine the extension (variance) of the point cloud along the axes defined by the eigenvectors. The spectral coefficients of a patch are shown as a signal for the x axis in Fig. 3(a) and as a 3D distribution in Fig. 3(b).

The watermark embedding method has three steps. Firstly, the point cloud of spectral coefficients $\mathbf{C}_i$ is rotated such that its axes coincide with the orthogonal axes defined by the eigenvectors:

$$\mathbf{D}_i = \mathbf{C}_i \mathbf{U}_i \tag{17}$$

In this case the variances along the three axes are not correlated with each other.

The cloud of 3-D points of $\mathbf{C}_i$ is then "squashed" for embedding a bit of 1 and "inflated" to a sphere for embedding a bit of 0, by using the ratio between the eigenvalues :

$$\frac{\lambda_1}{\lambda_k} = K \quad \begin{cases} K > 1 \text{ for a bit of 1} \\ K = 1 \text{ for a bit of 0} \end{cases} \tag{18}$$

where $k \in \{2, 3\}$. In order to enforce these constraints, the variance along the second and third axis is changed without affecting the variance corresponding to the largest eigenvalue :

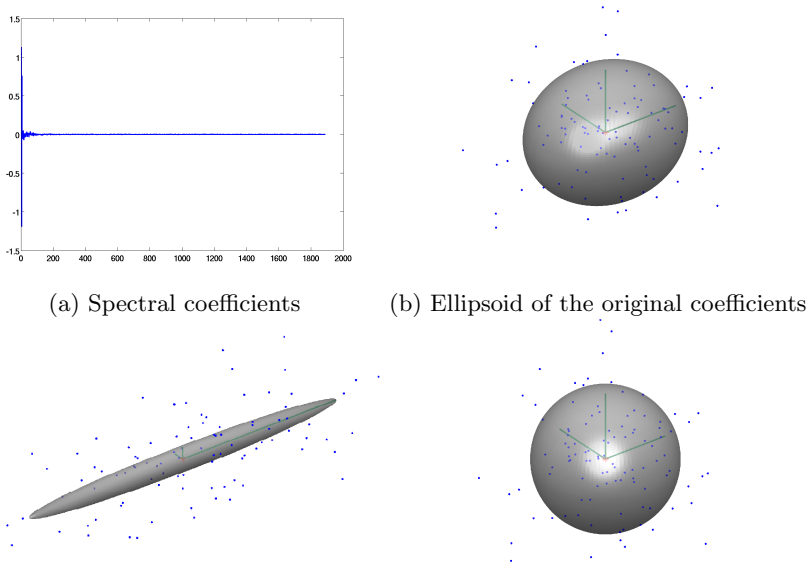
$$\hat{D}_{i,k} = D_{i,k} \sqrt{\frac{\lambda_1}{K \lambda_k}} \tag{19}$$

where $k \in \{2, 3\}$, λ_1 is the highest variance, corresponding to the cloud principal axis, and $\hat{D}_{i,k}$ represents the modified k component of the coefficient vector after embedding the watermark. For embedding a '1' bit, K is set to be larger than 1, and for embedding a '0' bit, K is set to be 1. Figs. 3(c) and 3(d), illustrate the shape of the coefficients cloud after embedding a bit of 0 and 1, respectively.

The watermarked spectral coefficients of high frequency are reconstructed as:

$$\hat{\mathbf{C}}_i = \hat{\mathbf{D}}_i \mathbf{U}_i^T \tag{20}$$

Finally, we enforce the changes back to the watermarked coefficients $\hat{\mathbf{C}}_i$. The reverse transformation of equation (13), using the watermarked coefficients $\hat{\mathbf{C}}_i$, is used for recovering the geometry of each individual patch. The entire watermarked graphical object is reconstructed by connecting back the patches with each other in the reversal of the procedure from Section 3.



(a) Spectral coefficients (b) Ellipsoid of the original coefficients

(c) Ellipsoid with a bit of 1 embedded (d) Ellipsoid with a bit of 0 embedded

Fig. 3. Enforcing constraints into spectral coefficients of meshes.

5.2 Watermark extraction

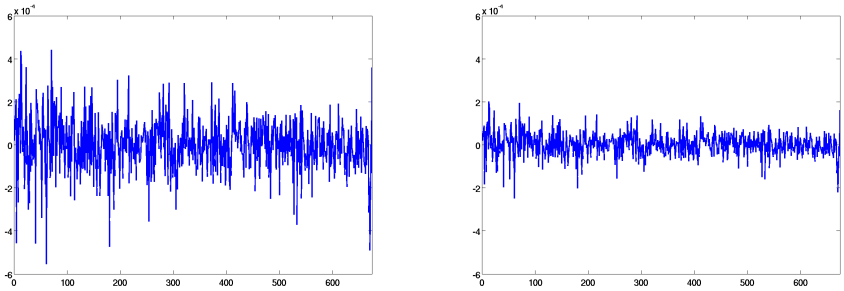
The proposed spectral PCA watermarking detection stage does not require the original object for retrieving the watermark. First of all, the mesh object is aligned and segmented as explained in Sections 2 and 3 Then, we apply the spectral analysis on each patch and extract the spectral coefficients in the same way as proposed in Section 4. Finally, we calculate the ratio between the largest and the smallest eigenvalues of the point cloud formed by the watermarked coefficients and retrieve the information bit as :

$$\begin{array}{ll} \text{if } \frac{\hat{\lambda}_1}{\hat{\lambda}_3} > T & \text{then } bit = 1 \\ \text{otherwise} & \text{then } bit = 0 \end{array} \tag{21}$$

where T is a threshold which depends on the embedding level K.

6 Experimental results

The proposed 3D watermarking algorithm is applied on four different mesh objects: Bunny with $34,835$ vertices and $69,666$ faces, Horse with $67,583$ vertices and $135,162$ faces, Buddha with $89,544$ vertices and $179,222$ faces and Venus head with $134,345$ vertices and $268,686$ faces. Each object is split into $N = 70$ patches grouped into $\kappa = 2$ layers as described in Section 3 while embedding a total of $M = 64$ bits. It can be observed that these objects contain many vertices and faces but by splitting them into patches as explained in Section 3 we reduce the required computational complexity for spectral watermarking. The watermark algorithm parameters are set as: $\alpha = 0.1$, $K = 15$ and $T = 2.25$, as



(a) Original (b) After embedding a bit of '1'

Fig. 4. Spectral coefficients corresponding to the y axis component.

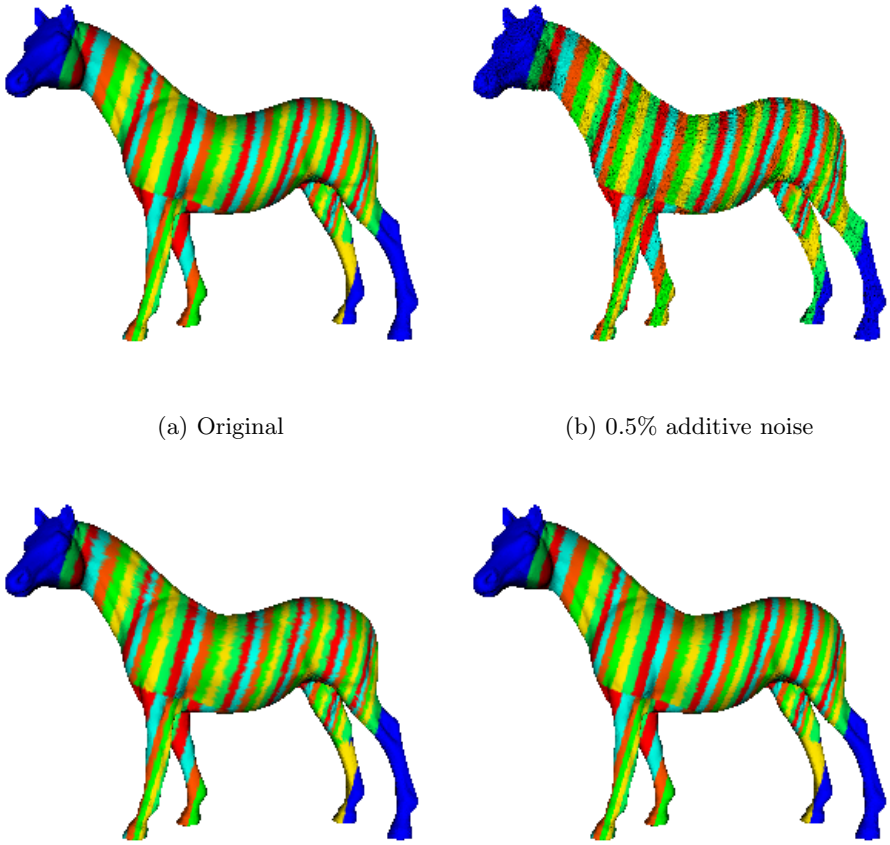
used in equations (5), (18) and (21), respectively. The spectral coefficients corresponding to the y axis coordinate, *i.e.* corresponding to the second component eigen-vector, before and after watermarking are shown in Figs. 4(a) and 4(b), respectively. It can be observed that the amplitude of the high frequency coefficients is shrunk after watermarking a bit of '1'. High frequency modifications only introduces small geometric distortions to the shape.

In Fig. 5 we experimentally examine the robustness of the equal area segmentation method proposed in Section 3. The segmentation result on the original object is shown in Fig. 5(a). The results obtained after considering additive noise, mesh simplification and Laplacian smoothing are shown in Figs. 5(b), 5(c) and 5(d), respectively. For the simplification, we used the quadric error metric software described in [14], while for the mesh smoothing we employed the Laplacian filter proposed in [15] with a parameter $\lambda = 0.2$ and for 10 iterations. The segmentation is consistent almost perfectly under the simplification and the Laplacian smoothing. Some errors emerge at the leg of the horse under the additive noise attack however most of the segments remain identical with the original ones.

The proposed 3D graphics watermarking algorithm is compared with the state-of-art robust algorithm proposed by Cho *et al.* in [3]. We denote by ChoMean and ChoVar, the mean change and the variance change algorithms, described in [3]. We set the watermark parameter $\alpha = 0.05$ for the ChoMean and ChoVar methods according to their embedding algorithm from [3].

The distortion introduced by our spectral watermarking algorithm is compared objectively and visually. As the numerical objective comparison measure, we use the MRMS proposed in [16]. The comparison of the visual distortions is shown in Fig. 6, while the numerical results are listed in Table 1. From these results it is clear that the algorithm proposed in this paper introduces less distortion than Cho's algorithms for all four objects from both geometric and visual points of view.

The robustness comparison results of the three algorithms against various attacks such as additive noise, mesh simplification, quantization and Laplacian smoothing are shown in Figs. 7 and 8, respectively. For smoothing, the robustness results are basically similar for all three methods; our algorithm produces slightly

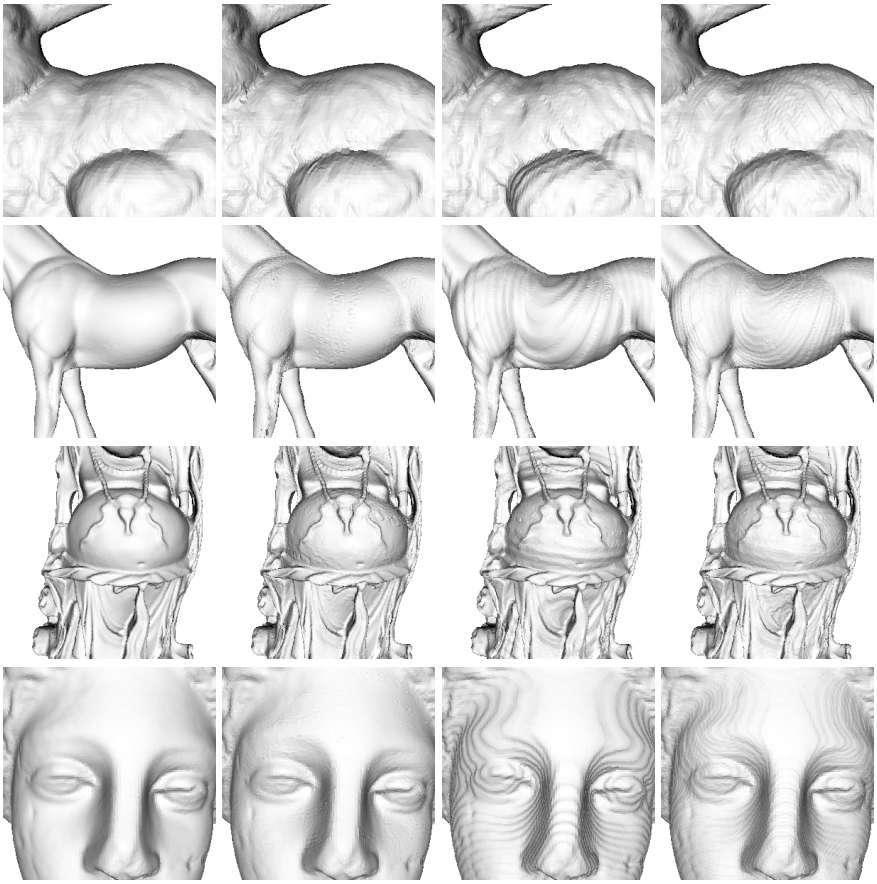


(a) Original (b) 0.5% additive noise

(c) 50% simplification (d) Smoothed after 10 iterations, $\lambda = 0.2$.

Fig. 5. Robustness of the segmentation method.

better results when using fewer smoothing iterations. Our algorithm is also better than Cho's algorithms for quantization attacks. However, the Cho's results are better when considering mesh simplification. Finally for additive noise, the results are basically similar for all three methods for Buddha and Head objects, while Cho's algorithms demonstrate a higher robustness in the case of Bunny and Horse objects. This comparison demonstrates the good trade-off of the proposed method between the watermark robustness and the distortion. The proposed algorithm introduces a lower distortion, both geometrically and visually, when compared to the Cho's methods at the price of a lower robustness for certain attacks, while maintaining the same robustness for others.



(a) Original (b) Spectral (c) ChoMean (d) ChoVar

Fig. 6. Visual distortions introduced by different watermarking algorithms.

7 Conclusion

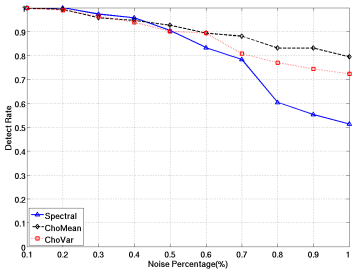
In this paper we propose a local blind spectral watermarking method for 3D meshes. The 3D object is split into patches in order to minimize the required computational complexity of the spectral decomposition. This segmentation algorithm produces equal area regions and is robust to both geometry and connectivity attacks. The watermark is then embedded by enforcing constraints on the spectral coefficients corresponding to each patch. The spectral coefficient distribution corresponds to an ellipsoid when embedding a bit of 1 and to a sphere when embedding a bit of 0. The proposed methodology provides increased watermark security and was shown to produce minimal distortion in the object shape, both from geometric and visual points of view. Extensive experiments have shown that the proposed method provides a good trade-off between distortion and robustness when compared with state-of-the-art spatial domain methods.

Object	Spectral	ChoMean	ChoVar
Bunny	1.91	4.95	2.29
Horse	0.43	0.87	0.54
Buddha	0.39	0.78	0.47
Head	0.10	0.25	0.15

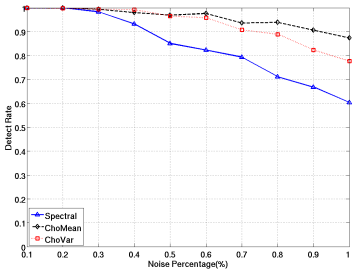
Table 1. Geometric distortions measured by MRMS ($\times 10^{-4}$).

References

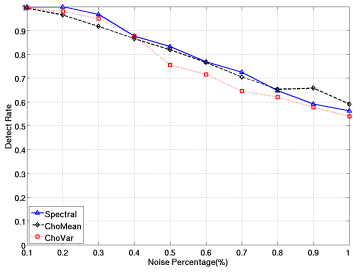
1. Bors, A.G.: Watermarking mesh-based representations of 3-D objects using local moments. IEEE Trans. on Image Processing **15**(3) (2006) 687–701 1
2. Zafeiriou, S., Tefas, A., Pitas, I.: Blind robust watermarking schemes for copyright protection of 3D mesh objects. IEEE Trans. on Visualization and Computer Graphics **11**(5) (2005) 496–607 1
3. Cho, J.W., Prost, R., Jung, H.Y.: An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. IEEE Trans. on Signal Processing, 2007 **55**(1) 142–155 1, 10
4. Ohbuchi, R., Mukaiyama, A., Takahashi, S.: A frequency-domain approach to watermarking 3-D shapes. In: Proc. of Eurographics, Computer Graphics Forum, vol. 21. (2002) 373–382 1
5. 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 Processing: Image Communication **18**(4) (2003) 309–319 1
6. Wang, K., Lavoué, G., Denis, F., Baskurt, A.: A comprehensive survey on three-dimensional mesh watermarking. IEEE Trans. on Multimedia **10**(8) (2008) 1513–1527 1
7. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proc. of SIGGRAPH, Computer Graphics Proceedings. (2000) 279–286 1
8. Lavoué, G., Denis, F., Dupont, F.: Subdivision surface watermarking. Computers & Graphics **31**(3) (2007) 480–492 1
9. Cotting, A., Weyrich, T., Pauly, M., , Gross, M.: Robust watermarking of point-sampled geometry. In: Proc. of Int. Conf. on Shape Modeling and Applications. (2004) 233–242 1
10. Liu, Y., Prabhakaran, B., Guo, X.: A robust spectral approach for blind watermarking of manifold surfaces. In: Proc. of ACM Multimedia and Security Workshop. (2008) 43–52 1
11. Luo, M., Bors, A.G.: Principal component analysis of spectral coefficients for mesh watermarking. In: Proc. of IEEE Int. Conf. on Image Processing. (2008) 441–444 1, 2, 3
12. Zhang, C., Chen, T.: Efficient feature extraction for 2D/3D objects in mesh representation. In: Proc. of IEEE Int. Conf. on Image Processing. (2001) 935–938 2, 3
13. Tuzikov, A., Sheynin, S., Vasiliev, P.: Computation of volume and surface body moments. Pattern Recognition **36**(11) (2003) 2521–2529 3
14. Garland, M., Heckbert, P.: Surface simplification using quadric error metrics. In: Proc. SIGGRAPH, Graphical Models 66(6). (1997) 370–397 10
15. Taubin, G.: Geometric signal processing on polygonal meshes. In: Eurographics state of the art report. (2000) 10
16. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. Computer Graphics Forum **17**(2) (1998) 167–174 10



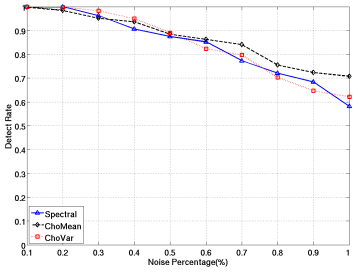
(a) Bunny



(b) Horse

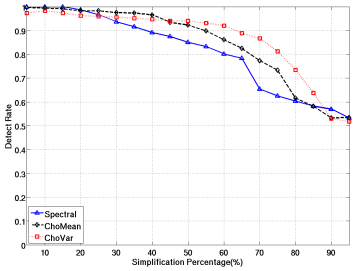


(c) Buddha

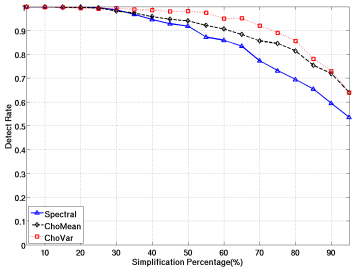


(d) Head

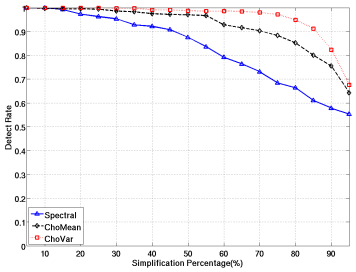
Robustness to additive noise



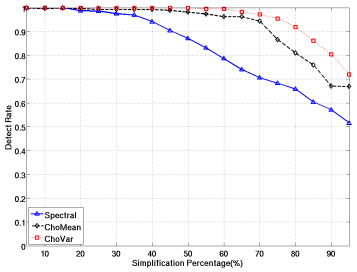
(e) Bunny



(f) Horse



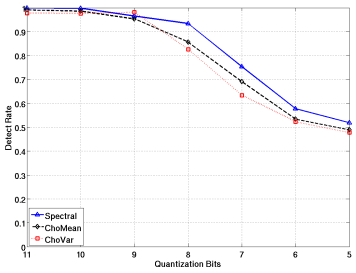
(g) Buddha



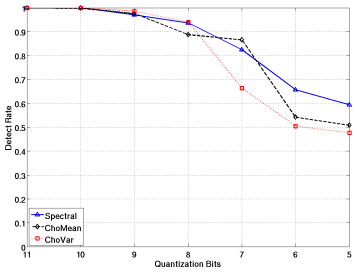
(h) Head

Robustness to mesh simplification

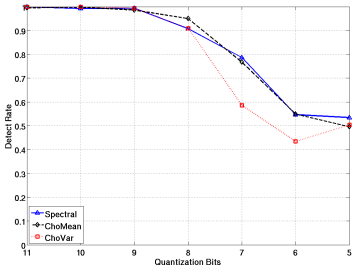
Fig. 7. Robustness to additive noise and mesh simplification.



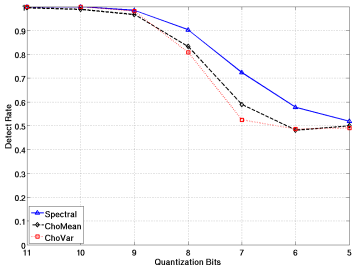
(a) Bunny



(b) Horse

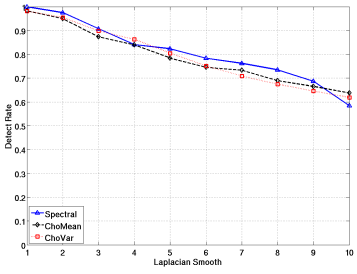


(c) Buddha

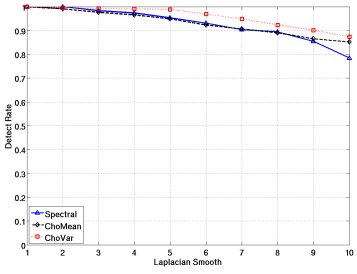


(d) Head

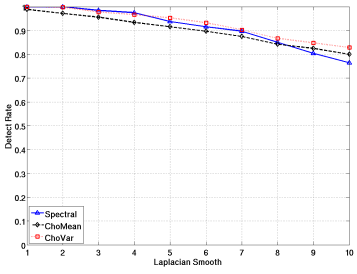
Robustness to bit quantization



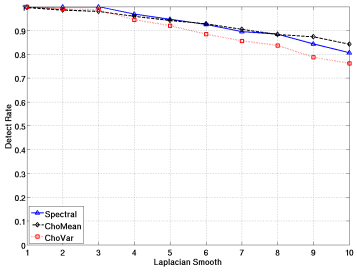
(a) Bunny



(b) Horse



(c) Buddha



(d) Head

Robustness to mesh smoothing

Fig. 8. Robustness to quantization and mesh smoothing.